**BOOK BANK SYSTEM**

**AIM:**To create a system to perform book bank operation

# PROBLEM STATEMENT:

A Book Bank lends books and magazines to member, who is registered in the system. Also it handles the purchase of new titles for the Book Bank. Popular titles are brought into multiple copies. Old books and magazines are removed when they are out or date or poor in condition. A member can reserve a book or magazine that is not currently available in the book bank, so that when it is returned or purchased by the book bank, that person is notified. The book bank can easily create, replace and delete information about the tiles, members, loans and reservations from the system.

# SOFTWARE REQUIREMENTS SPECIFICATION:

**INTRODUCTION**

Book Bank is the interface between the students and Librarian. It aims at improving the efficiency in the Issue of books or magazines and reduce the complexities involved in it to the maximum possible extent.

# PURPOSE

If the entire process of 'Issue of Books or Magazines' is done in a manual manner then it would take several months for the books or magazines to reach the applicant. Considering the fact that the number of students for Book Bank is increasing every year, an Automated System becomes essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process. The system has been carefully verified and validated in order to satisfy it.

# SCOPE

The System provides an online interface to the user where they can fill in their personal details and submit the necessary documents (may be by scanning). The authority concerned with the issue of books can use this system to reduce his workload and process the application in a speedy manner.

# DEFINITIONS, ACRONYMS AND THE ABBREVIATIONS

* + - **Librarian** - Refers to the super user who is the Central Authority who has been vested with the privilege to manage the entire system.
    - **Student** - One who wishes to obtain the Books or Magazines.
    - **HTML** - Markup Language used for creating web pages.

# TOOLS TO BE USED

* + - StarUML

# OVERVIEW

SRS includes two sections overall description and specific requirements.

**Overall description** will describe major role of the system components and inter- connections.

**Specific requirements** will describe roles & functions of the actors.

# OVERALL DESCRIPTION:

It will describe major role of the system components and inter-connections.

# PRODUCT PERSPECTIVE

The SRS acts as an interface between the 'Students' and the 'Librarian'. This system tries to make the interface as simple as possible and at the same time not risking the security of data stored in. This minimizes the time duration in which the user receives the books or magazines.

# SYSTEM FUNCTIONS

* Secure Registration of information by the Students.
* Librarian can generate reports from the information and is the only authorized personnel to add the eligible application information to the database.

# USER CHARACTERISTICS

* **Student** - They are the people who desire to obtain the books and submit the information to the database.
* **Librarian** - He has the certain privileges to add the books and to approval of the reservation of books.

# CONSTRAINTS

* The Students require a computer to submit their information.
* Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring.
* The Students has to be careful while submitting the information. Much care is required.

# ASSUMPTIONS AND DEPENDENCIES

* The Student and Librarian must have basic knowledge of computers and English Language.
* The Students may be required to scan the documents and send.

# USE-CASE DIAGRAM:

The book bank use cases are:

* 1. book\_issue
  2. book\_return
  3. book\_order
  4. book\_entry
  5. search book\_details

# ACTORS INVOLVED:

1. Student
2. Librarian
3. Vendor

# USECASE NAME : SEARCH BOOK\_DETAILS

The librarian initiates this use case when any member returns or request the book and checking if the book is available.

**Precondition:** The librarian should enter all Book details.

**Normal Flow:** Build message for librarian who search the book.

**Post Condition:** Send message to respective member who reserved the book.

# Usecase Name : Book\_Issue

Initiated by librarian when any member wants to borrow the desired book. If the book is available, the book is issued.

**Precondition:** Member should be valid member of library.

**Normal Flow:** Selected book will be issued to the member.

**Alternative Flow:** If book is not available then reserved book use case should be initiate.

**Post Condition:** Update the catalogue.

# Usecase Name : Book\_Order

Initiated by librarian when the requested book is not available in the library at that moment. The book is reserved for the future and issued to the person when it is available.

**Precondition:** Initiated only when book is not available.

**Normal Flow:** It reserved the book if requested.

**Post Condition :** Mention the entry in catalogue for reservation.

# USECASE NAME : BOOK\_RETURN

Invoked by the librarian when a member returns the book.

**Precondition:** Member should be valid member of library.

**Normal Flow:** Librarian enters bookid and system checks for return date of the book. **Alternative Flow:** System checks for return date and if it returned late fine message will be displayed.

**Post Condition: C**heck the status of reservation.

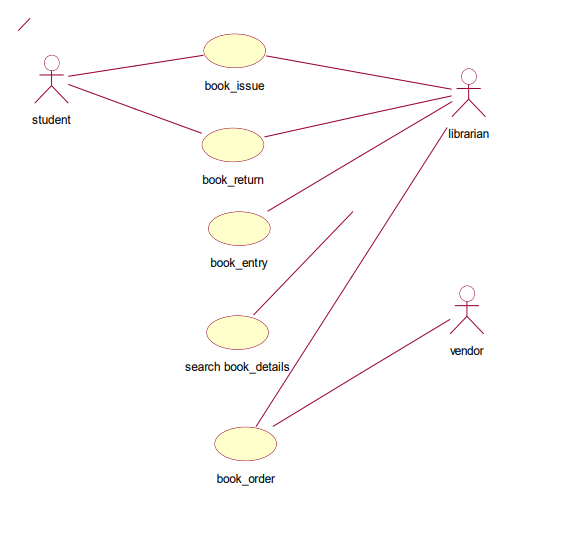
# USECASE NAME : BOOK\_ENTRY

The purchase book use-case when new books invoke it or magazines are added to the library.

**Precondition:** Not available or more copies are required.

**Normal Flow:** Enter bookid,author information, publication information, purchased date, prize and number of copies.

**Post Condition:** Update the information in catalogue.



# FIG:USE-CASE DIAGRAM FOR BOOK BANK SYSTEM

**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. An activity is shown as an rounded box containing the name of the operation.

This activity diagram describes the behaviour of the system.

shows id card



request for specific book

is book available? no

enquires for alternative book

yes

borrows book

yes

if satisfied?

no

librarian approves transaction

transaction cancelled

# ACTIVITY DIAGRAM [BORROW BOOK]



collects quotation from vendors

no

if satisfied with norms?

yes

place order

discount & mode of payment finalised

takes delivery

bill amt paid

**ACTIVITY DIAGRAM [ ORDER BOOK]**

**ACTIVITY DIAGRAM [RETURN BOOK]**



shows id and library card

librarian makes entry

on or before return date?

no

pays fine

yes

librarian approves transaction

**CLASS DIAGRAM:**

The class diagram, also referred to as object modeling is the main static analysis diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects.

The ATM system class diagram consists of four classes:

* 1. Student
  2. Book
  3. Issue
  4. Return
  5. Vendor
  6. Details

# STUDENT:

It consists of twelve attributes and three operations. The attributes are enrollno, name, DOB, fathername, address, dept name, batch and book limits. The operations of this class are addStInfo(), deleteStInfo(), modifyStInfo().

# BOOK:

It consists of ten attributes and four operations. This class is used to keep book information such as author, title, vendor, price, etc

# ISSUE:

It consists of eight attributes and two operations to maintain issue details such as, issue date, accno of issued book, name of the student who borrowed book.

# RETURN:

It consists of eight attributes and two operations to maintain issue details such as, issue date, accno of issued book, name of the student who borrowed book.

# STUDENTS:

The attributes of this class are name, dept ,year ,bcode no The operation is display students().

# DETAIL:

The attributes of this class are book name, author, bcode no The operations are delete details().



# CLASS DIAGRAM FOR BOOK BANK SYSTEM

**SEQUENCE DIAGRAM:**

A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices.

An event also is considered to be any action by an object that sends information. The event line represents a message sent from one object to another, in which the “form” object is requesting an operation be performed by the “to” object. The “to” object performs the operation using a method that the class contains.

It is also represented by the order in which things occur and how the objects in the system send message to one another.

**: student : librarian**

**: issue**

**: return**

**search**

**DB**

**DB**

**search**

**: return**

**: issue**

**7**

**1: request book**

**2: check available book**

**check available book**

**4: not avilable**

**: request for another book**

**9: check availabilty**

**10: available**

**13: provide student details**

**14: enter issue data**

**15:**

**ly**

**19: request to return book**

**20: enter the book details**

**21: update return status**

**atus updated**

**22: return st**

**d successfully**

**23: update**

**24: book returned**

**d**

**ssue status update**

**16: i**

**: updated successful**

**18: issue book**

**17**

**update issue status**

**11: avilable**

**12: avilable**

**8: check availability**

**3:**

**5: not avilable**

**6: not available**

# SEQUENCE DIAGRAM FOR BOOK ISSUE & RETURN

**1: request book**

**7: request for another book 13: provide student details 19: request to return book**



**6: not available**

**12: avilable**

**18: issue book**

**: librarian**

**14: enter issue data**

**17: updated successfully**

**: issue**

**15: update issue status**

**: student**

**24: book returned**

**16: issue status updated**

**23: updated successfully**

**20: enter the book details**

**2: check available book 8: check availability**

**5: not avilable**

**11: avilable**

**DB**

**3: check available book**

**21: update return status**

**9: check availabilty**

**22: return status updated**

**4: not avilable**

**10: available**

**search**

**: return**

# COLLABORATION DIAGRAM FOR BOOK ISSUE & RETURN

**STATE CHART DIAGRAM**

It consists of state, events and activities. State diagrams are a familiar technique to describe the behavior of a system. They describe all of the possible states that a particular object can get into and how the object's state changes as a result of events that reach the object



start

book issue

book return

vendor supplies book to the library

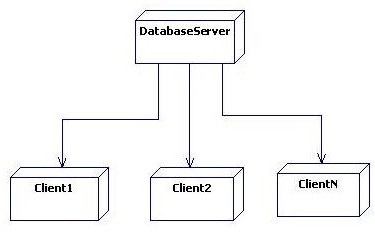
student takes book from library

student returns book to library[ pays fine if needed]

# STATE CHART DIAGRAM

**DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM**

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.



# DEPLOYMENT DIAGRAM