Id: b180305047

1. **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.

1. **Preparation of Software Requirement Specification Document:**

* **Users 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.
* **System Modules:**
* **Log in:** Secure registration of student and librarian by filling online registration form.
* **Book bank:** Book bank contains all the books. New book added to the book bank with bookno, title name, author, edition, publisher name details to the database. Any book is deleted if damaged. Update of the book information also done.
* **Operations:** Student and administrator perform their operations like add book, delete book, update information, view book details are implemented in log in Web Pages.
* **Non-functional requirements:**
* **Privacy:** Privacy maintained for each and every user by providing user credentials username and password.
* **Portability:** Installation on multiple platforms and execution of software.

1. **Preparation of Software, Hardware Configuration Management:**

* **Software Requirements:**

Operating system: windows 7/10

Front end: J2EE

Back end: My SQL Server

IDE used: Netbeans

* **Hardware Requirements**:

Processor: i3 or higher

RAM: 4 GB

Hard Disk drive: 500 GB

1. **Study and usage of any Design phase CASE tool:**

* **CASE Tool:** **Draw.io**

Draw.io is a **UML** (Unified Modeling Language) tool. It is an open-source modeling tool that supports the UML framework for system and software modeling. Draw.io provides different types of diagram. It released for beta testing under a property license. Draw.io is actively supporting the **MDA** (Model Driven Architecture). It approaches by supporting the UML profile concept and allowing it to generate code for multiple languages. It also provides a number of bug fixes and improved compatibility with the modern versions of the Windows Operating System. Draw.io is mostly used by the Agile and small development teams, professional persons and used by the educational institutes.

* **Diagram Types in Draw.io:**

1. Use Case Diagram

2. Class Diagram

3. Sequence Diagram

4. Collaboration Diagram

5. State chart Diagram

6. Component Diagram

7. Deployment Diagram

8. Composite Structure Diagram

1. **Performing the Design by using any Design phase CASE tools:**

**CASE Tool:** **Draw.io**

* **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:** Check 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.

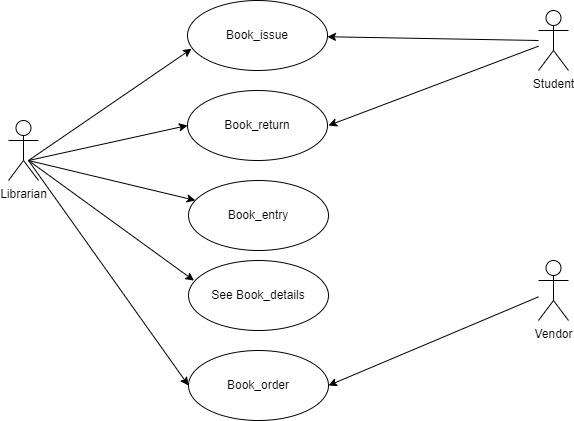


Figure . Usecase diagram for Book Bank System

* **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.

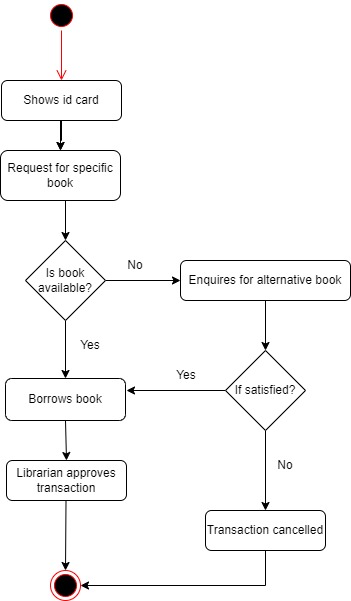


Figure . Activity Diagram for Book Bank System

* **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.

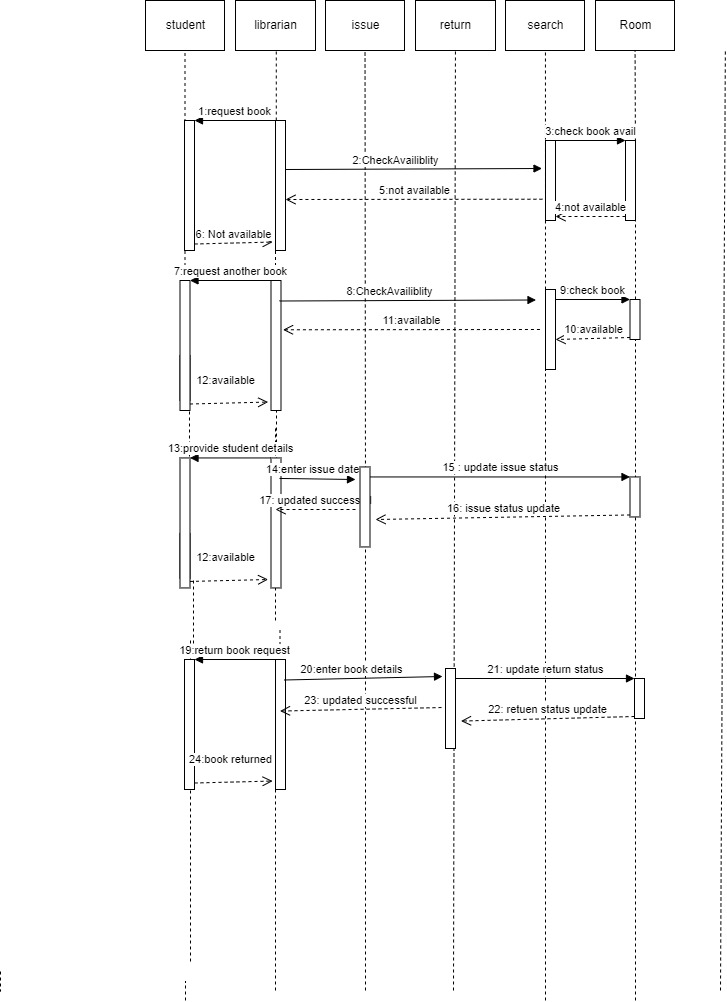


Figure . Sequence Diagram For Book Issue

* **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

**1) 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().

**2) Book:**

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

**3) 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.

**4) 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.

**5) Students:**

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

**6) Detail:**

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

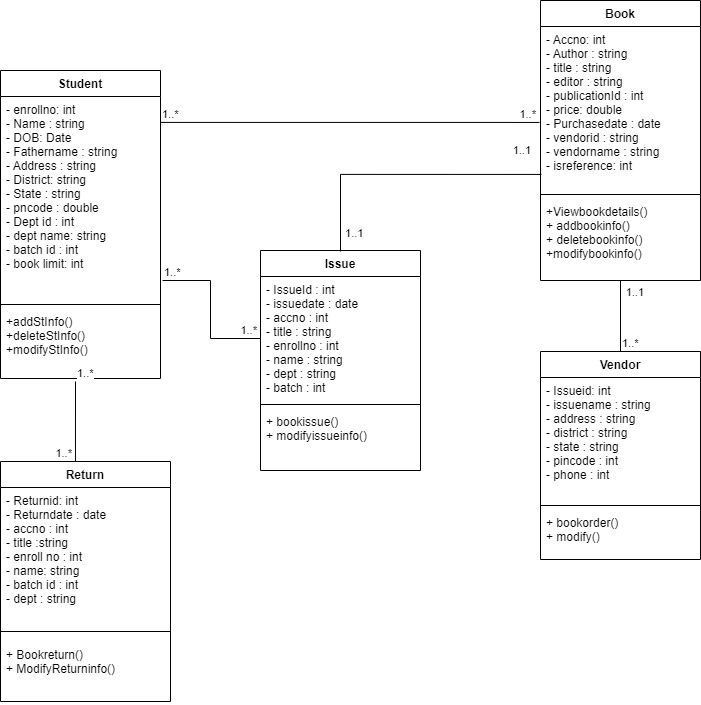


Figure . Class Diagram For Book Bank System

* **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.

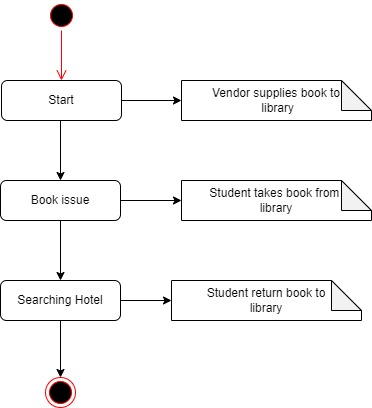


Figure . State Chart Diagram for BookBank System

* **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.

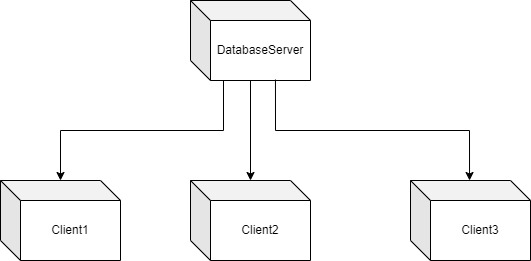


Figure : Deployment Diagram for Book Bank System