# Analysis

# Of

# Library management system

# Requirement Specification:-

**Requirement 1:**

**Description:** Add a student record.

- Each student should have following attributes:
  - Student ID/SSN (unique)
  - Name
  - Address
  - Phone number

**Requirement 2:**

**Description:** Update/ Delete a student record.

- The record would be selected using the student ID
- Updates can be made on full. Items only:
  - Name
  - Address
  - Phone number
- The record can be deleted if there are no books issued by user.

**Requirement 3:**

**Description:** Add a book item

- Each book shall have following attributes:
  - Call Number
  - ISBN
  - Title
  - Author name
- The call number format to be adopted would be as follows
  - (First 2 alphabets of title) –(first 2 alphabets of author) – sequence number of book
- The ISBN number format to be adopted would be as follows:
  - (Numeric code for subject)-(sequence number)
  - NNN-NNNNN
  - It would be unique for each specific title/author combination
- Author name shall support 30 character names
- Title shall support 100 character names

**Requirement 4:**

**Description:** Update / Delete a book item

- The book item can be retrieved using the call number
- The data items which can be updated are:
  - ISBN
  - Title
  - Author name
- The book can be deleted only if no user has issued it.

**Requirement 5:**

**Description:** Query the book database

- The product shall let Librarian query books' detail information by their ISBN number or Author or Title.
- The search results would produce a list of books, which match the search parameters with following details:
  - Call number
  - ISBN number
  - Title
  - Author
- The display would also provide the number of copies which is available for issue
- The display shall provide a means to select one or more rows to a user-list
- A detailed view of each book should provide information about check-in/check out status, with the borrower's information.
- The search display will be restricted to 20 results per page and there would be means to navigate from sets of search results.
- The user can perform multiple searches before finally selecting a set of books for check in or checkout. These should be stored across searches.

**Requirement 6:**

**Description:** Check-out a book

- Librarians and member of the library can check out a book using its call number.
- The checkout can be initiated from a previous search operation where user has selected a set of books.

- The student ID who is issuing the book would be entered.
- The issue date would automatically reflect the current system date.
- The due date would automatically be stamped as 7 days from current date.

**Requirement 7:**

**Description:** Check-in a book

- Librarians and member of the library can check in a book using its call number.
- The check-in can be initiated from a previous search operation where user has selected a set of books.
- The return date would automatically reflect the current system date.
- Any late fees would be computed as difference between due date and return date at rate of one dollar a day.

**Requirement 8:**

**Description:** Display book detail

- This view would display details about a selected book from search operation.
- The details to be displayed are:
  - Call number
  - ISBN
  - Title
  - Author
  - Issue status (In library or checked out)
  - If book is checked out it would display
  - User ID & Name
  - Checkout date
  - Due date

**Requirement 9:**

**Description:** View student detail

- Librarians can select a user record for detailed view.
- The detail view should show:
  - User name, ID, Address & Phone number.
  - The books issued by user with issue date, due date, call number, title.
  - Late fees & Fines summary and total.

## Performance Requirements:

- The check in and check out system will respond to the user no more than 7 seconds.
- The search function will respond to the user no more than 10 seconds.

## Security requirements:

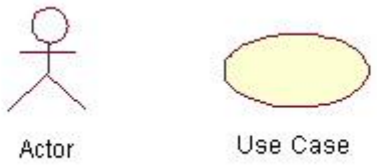
- All the account managing operations can be used by only librarian.
- The check-in, check-out and recall system can be used by only member of the library.
- The book return report and book borrowing report can be generated by only librarian.

## Reliability requirements:

- The system shall generate error messages when the user attempts to enter invalid data.
- If the system is down, it shall be recovered within 15 minutes.
- The system shall be recovered without interference at user terminal if it is down.
- The system shall have 99.9% reliability during library operating hours.

# Use Case Analysis:-

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.



An actor is represents a user or another system that will interact with the system you are modelling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

**Use Case Context diagram:-**

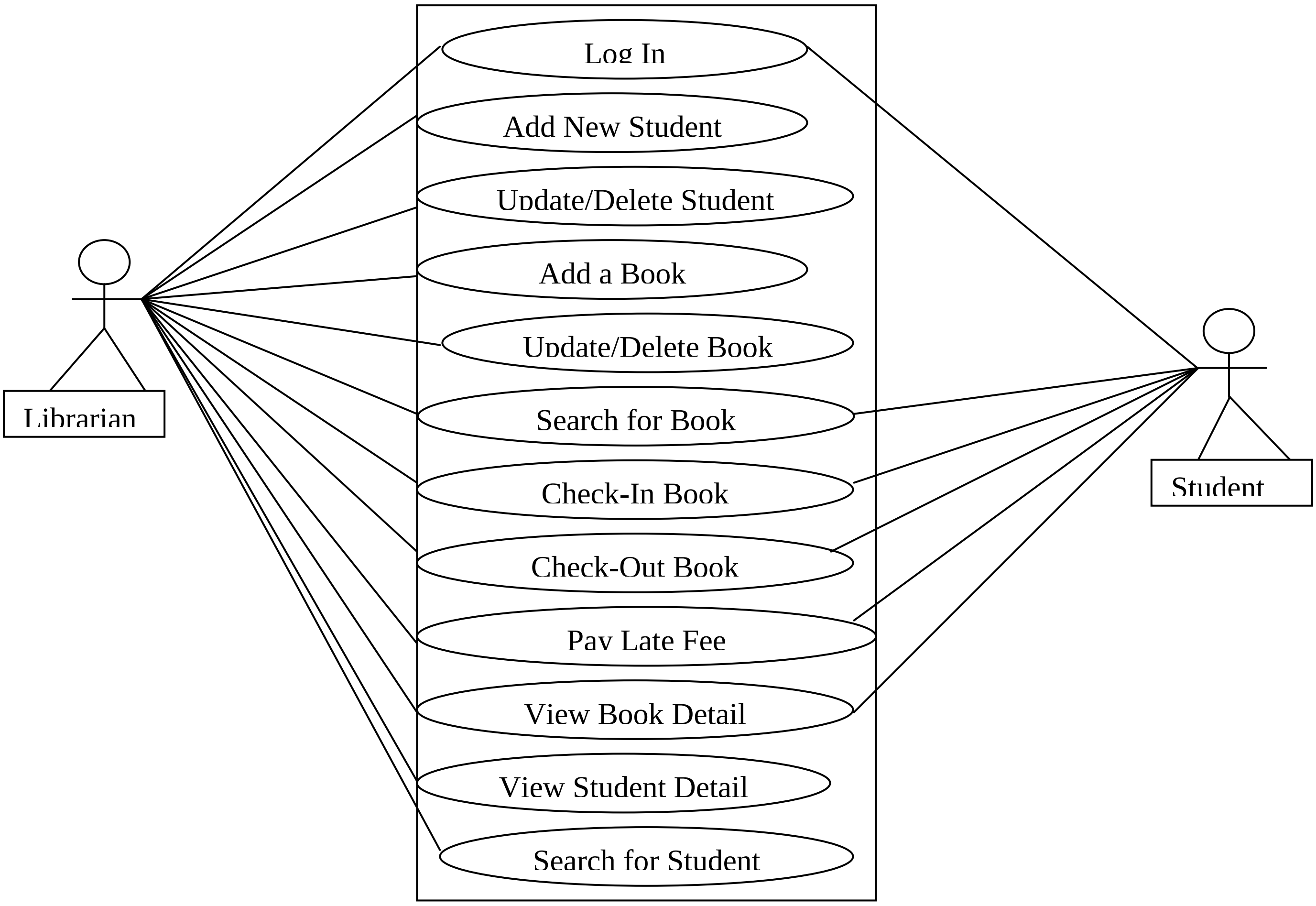


**Fig 1: Use case diagram of LMS (Library management system)**

The above diagram reveals the high-level functionality of the system.

Here in this system, the primary user of the system is librarian who is responsible for user creation, book item creation, check-in, checkout and all search operations. The student refers to an end-user of the system who is a member of the library and a student at the university.

Use case's description is as shown below:-

**Name:** Add/Edit/Delete Student record.
**Description:** Only librarians are responsible for adding/editing/deleting student record.
**Actors:** Librarian
**Trigger:** According to the need of librarian.
**Result:** The details of the student get updated.
**Essential process:** Identify user by login process.
Carry out manipulation task according to requirement.
Check student details are updated or not.

**Name:** Add/Edit/Delete Book item.
**Description:** Only librarians are responsible for adding/editing/deleting student record.
**Actors:** Librarian
**Trigger:** According to the requirement of librarian.
**Result:** The details of the book get updated.
**Essential process:** Identify user by login process.
Carry out manipulation task according to requirement.
Check book details are updated or not.

**Name:** Search for book.
**Description:** Librarians and Student are responsible for search book.
**Actors:** Librarian, Student
**Trigger:** Librarian or student searches book according to their need.
**Result:** Book details should be displayed.
**Essential process:** Identify Librarian or Student by login process.
Search for required book.
Check availability of book.
Display required book details.

**Name:** Check-Out Book.
**Description:** Librarians and Student are responsible for check-out book.
**Actors:** Librarian, Student
**Trigger:** Student or librarian requests for check-out book with the help of book call number and their id.
**Result:** Check-out confirmed.
**Essential process:** Identify Librarian or Student by login process.
Check whether check-out is completed or not.

**Name:** Check-In Book.
**Description:** Librarians and Student are responsible for check-in book.
**Actors:** Librarian, Student

**Trigger:** Student or librarian requests for check-in book with the help of book call number and their id.
**Result:** Check-in confirmed.
**Essential process:** Identify Librarian or Student by login process.
Check whether check-in is completed or not.
Overdue alert

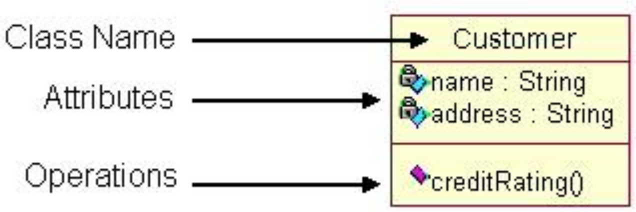
**Name:** Display book-detail.
**Description:** When the librarian or student searches book for borrow at that time the system shows the details of the book.
**Actors:** Librarian, Student
**Trigger:** Student or librarian searches book for borrowing.
**Result:** Specific book-details should be displayed in terms of call number, title, author etc...
**Essential process:** Identify Librarian or Student by login process.
Search a specific book.
Display book-detail.

**Name:** View student detail.
**Description:** Librarian can select a student record for detail view.
**Actors:** Librarian
**Trigger:** When the librarian wants to check student record.
**Result:** Specific student details displayed in terms of student id, phone number, books issued by that student with the issue date & due date, late fees etc…
**Essential process:** Identify Librarian or Student by login process.
Search a specific student.
View student detail.

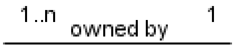
# Class diagram:-

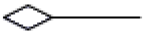
Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design.

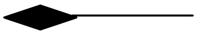
Classes are composed of three things: a name, attributes, and operations. Below is an example of a class.

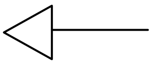


Association is a generic relationship between two classes and is modelled by a line connecting the two classes. This line also shows the feature multiplicity. (One-to-one, one-to-many, many-to many). For example



Aggregations indicate whole part-of relationship. It is represented by .

Composition relationship means the class is a member of another class. It cannot be present by itself. It is represented by 

Generalization relationship means “is-a “relationship. It is represented by 

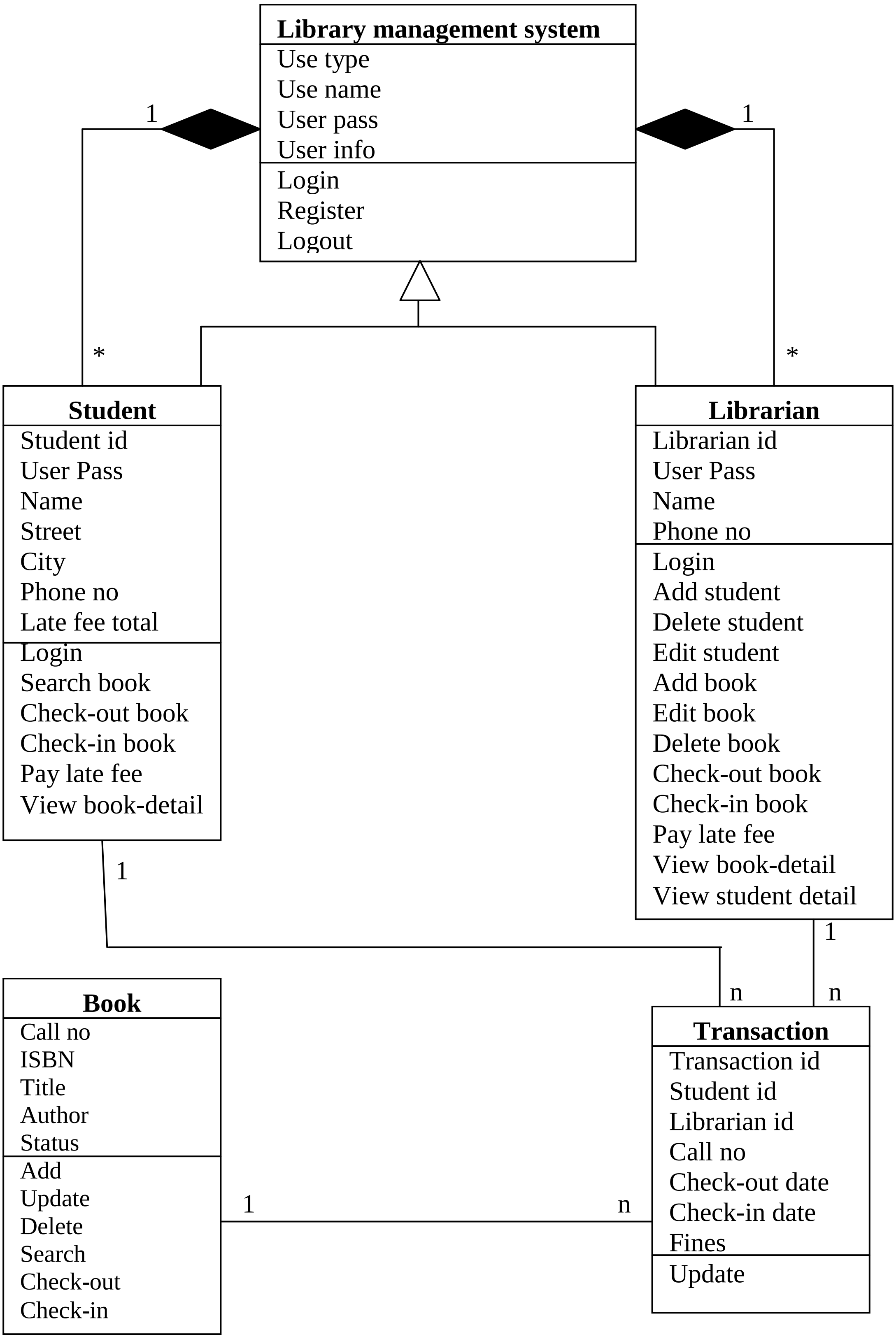


**Fig 2: Class diagram of LMS (Library management system)**

The above class diagram reveals the main class is library management system which is divided in to two users like student and librarian. Student and librarian perform tasks according to their need. The other class name is transaction and book.

# Activity diagram:-

Activity diagrams describe the workflow behaviour of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel.

**Activity states:** Activity states mark an action by object. It is represented by ⬭

**Transition**: When an activity states is completed processing moves to another activity state. Transition used to mark this movement. It is represented by →

**Initial state:** The initial state marks the entry point and the initial activity state. It is represented by ○

**Final state:** final states mark by the bulls' eye. It is represented by ◉

**Synchronization bar:** Activities often can be done in parallel. To spilt processing Synchronization bar is used. It is represented by ▬

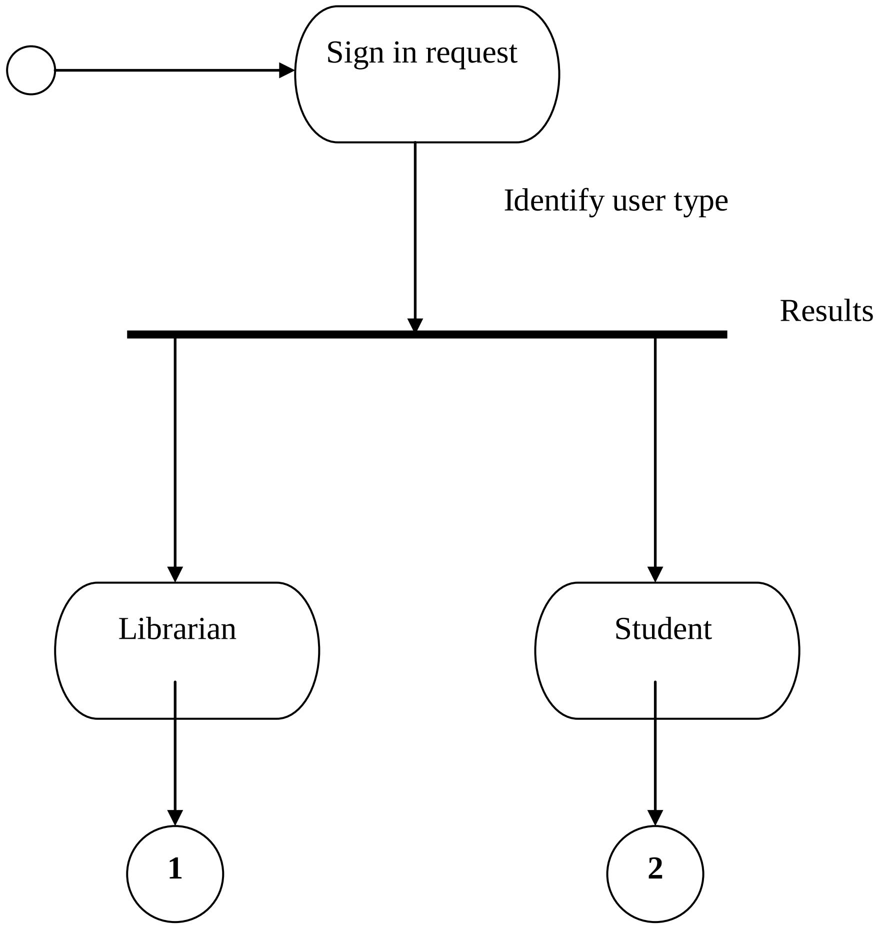


**Fig 3: Activity diagram of LMS (Library management system)**

The above diagram shows activity diagram of the library management system. In that, first user requests for sign in and system checks user type depending upon the data entered by user and displayed the results like librarian or student.

**Activity diagram of Librarian**

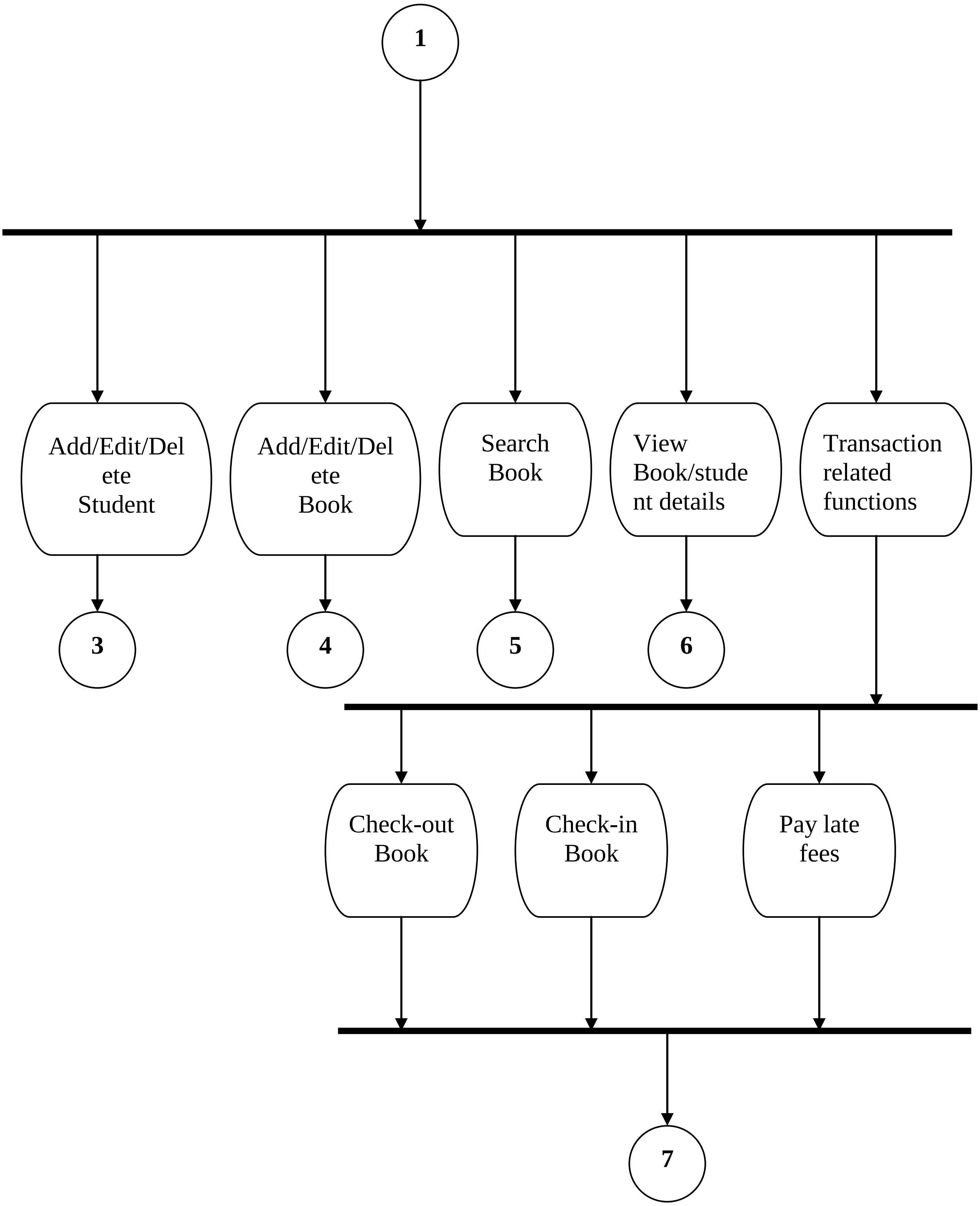


**Fig 4: Activity diagram of Librarian**

The above diagram shows the activity diagram of the librarian which includes activity performed by librarian. This is the primary user of the system that’s why he got full functionality of the system.

**End of the librarian’s activity**

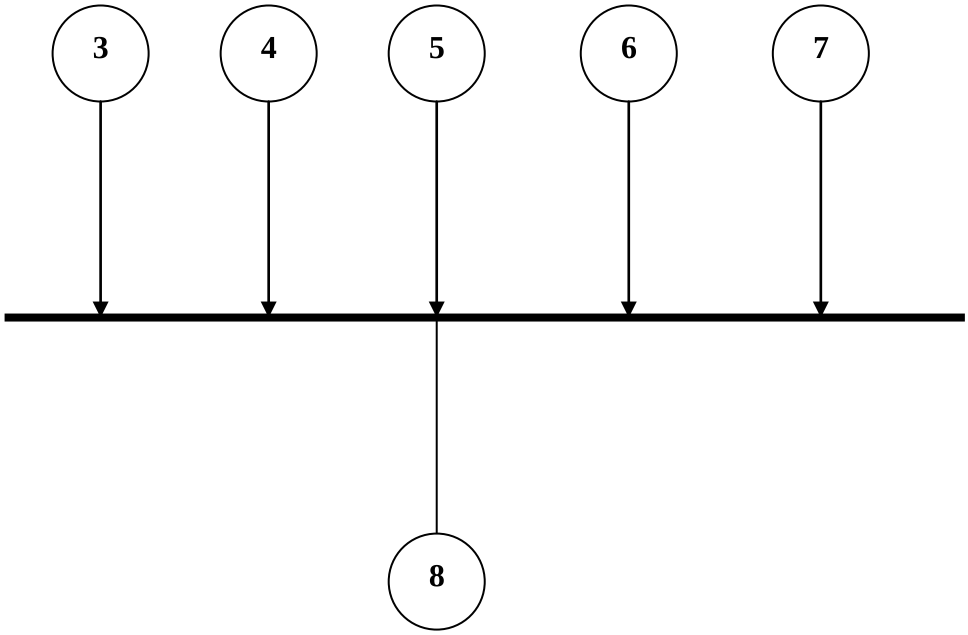


**Fig 5: End of the librarian’s activity**

The above diagram shows all individual activities of the librarian will be merging to the end of the activity which shows by number 8.

**Activity diagram of Student**

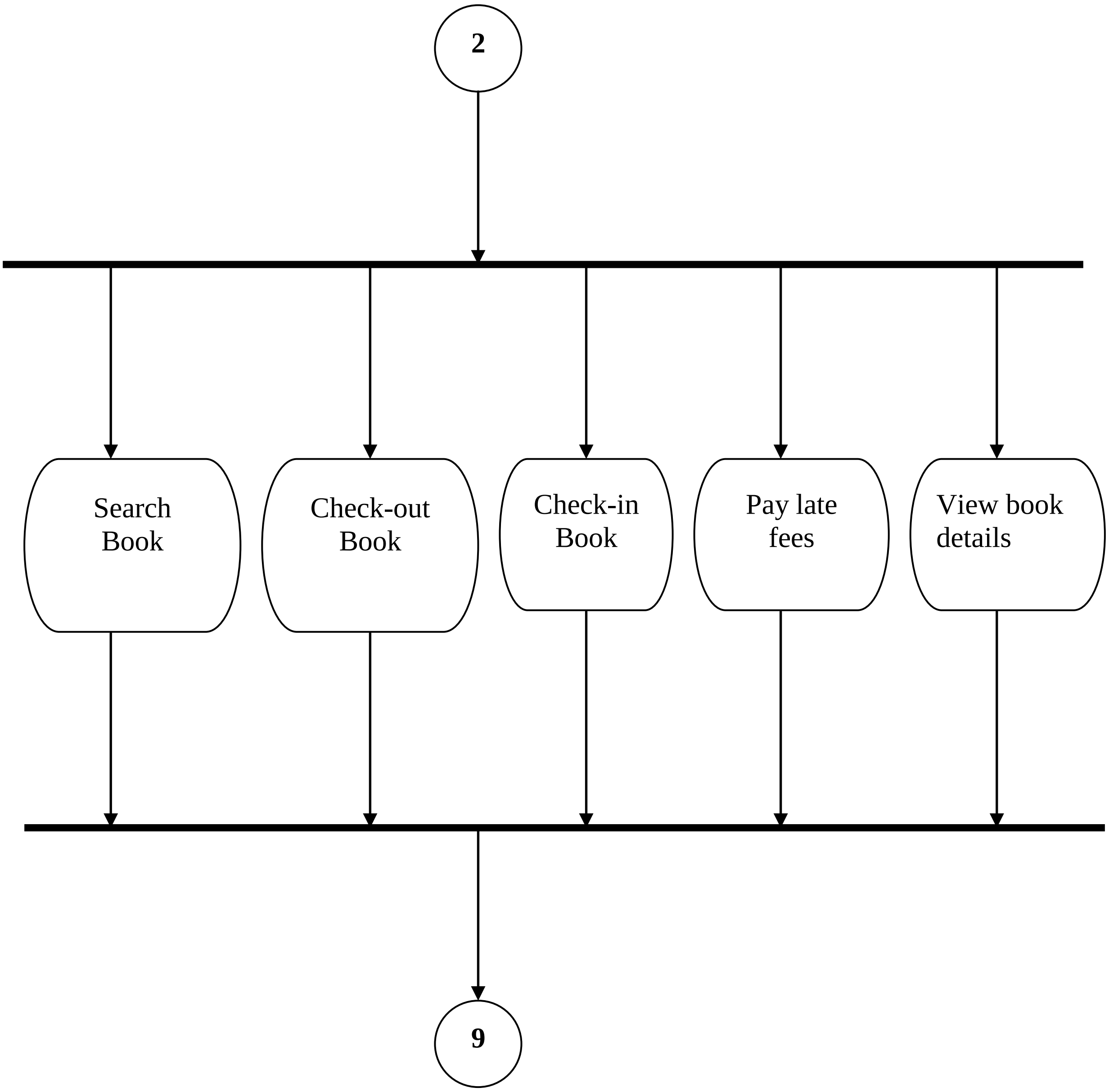


**Fig 6: Activity diagram of Student**

The above diagram shows the activity diagram of the student which includes activity performed by student. This is the end user of the system that's why he got less functionality of the system compare to the librarian.

**End of the Activity diagram of LMS (Library management system)**

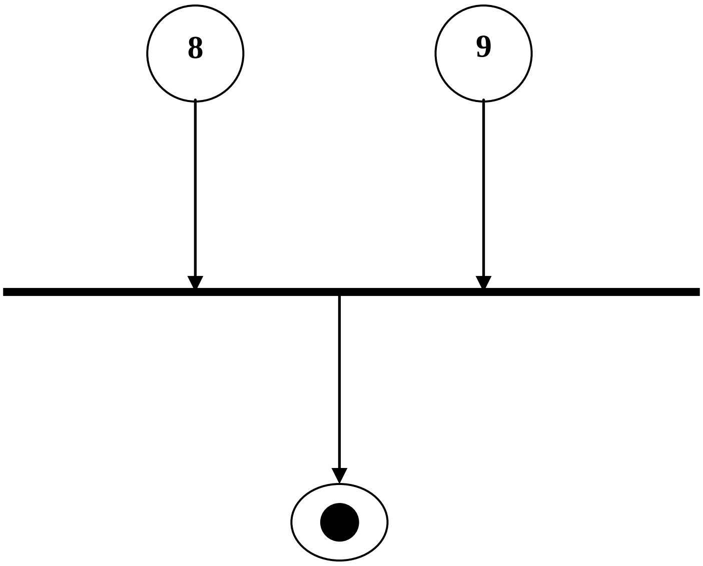


**Fig 7: End of the Activity diagram of LMS (Library management system)**

The above diagram shows all end activity of the librarian and student will be merging to the end of the activity of the system which shows by number 10.

# Component diagram:-

The component diagram contains components and dependencies. Components represent the physical packaging of a module of code. The dependencies between the components show how changes made to one component may affect the other components in the system. Dependencies in a component diagram are represented by a dashed line between two or more components. Component diagrams can also show the interfaces used by the components to communicate to each other.

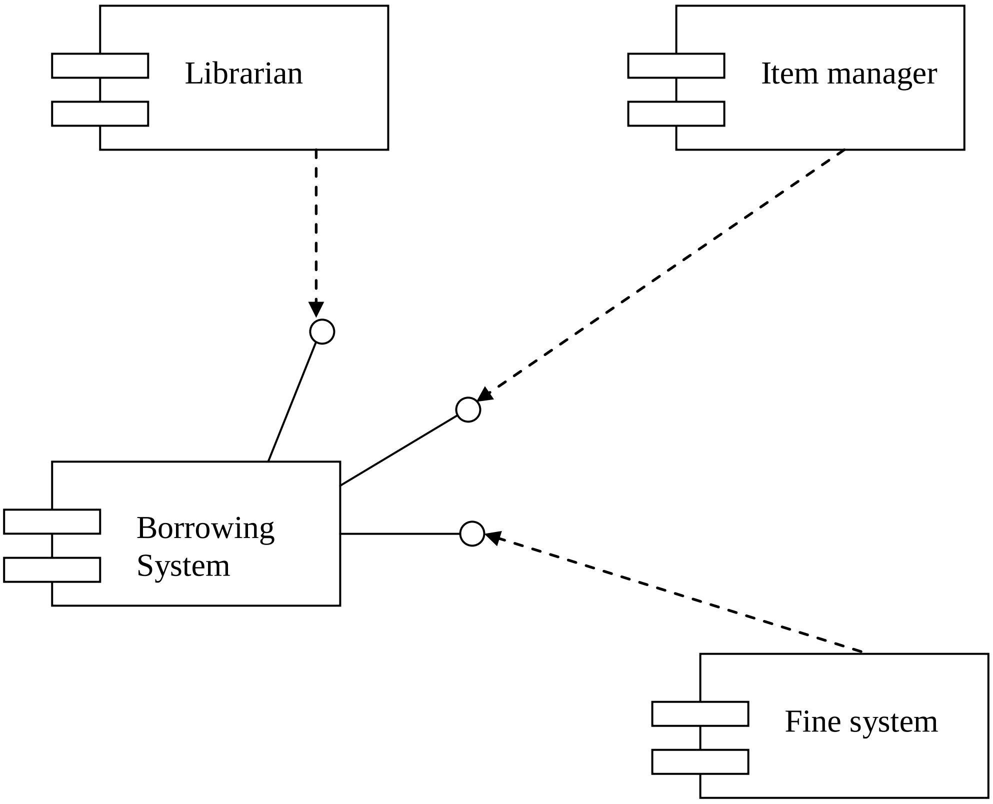


**Fig 8: Component diagram of LMS (Library management system)**

The above diagram shows four components of the library management system. In that, first component (librarian) is manager for the each student. Second component (fine system) that manages fines applied to the student who exceed the borrowing period. Third component (item manager) that manages library items. Fourth component (borrowing system) that manages all borrowing items.

# References

- http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/use_case.htm
- scotborders.g2b.biz/docs/schedule_of_requirement.doc
- www.cs.albany.edu/~xluo/high_designV1_1.doc
- ictlab.tyict.vtc.edu.hk/~billwang/OOT-Part-Time-CMM3332/Quiz2_Answer/Quiz2_answer.doc
- en.wikipedia.org/wiki/**Use_case_diagram**
- www.agilemodeling.com/artifacts/**classDiagram**.htm
- www.cs.albany.edu/~xluo/req_doc.doc
- en.wikipedia.org/wiki/Integrated**_library_system**
- www.dotnetcoders.com