

REGISTER NO:192371077

OOAD LAB

Ex>No:02	BOOK BANK REGISTRATION SYSTEM
Date:	

AIM:

To draw the diagrams [usecase, activity, sequence, collaboration, class, statechart, collaboration, component, deployment, package] for the Book bank registration system.

SOFTWARE REQUIREMENTS SPECIFICATION

SL.NO SOFTWARE REQUIREMENTS SPECIFICATION

1.0	Hardware Requirements
1.1	Software Requirements
1.2	Problem Analysis and Project Plan
1.3	Project Description
1.4	Reference

1.0 HARDWARE REQUIREMENTS:

Intel Pentium Processor I3/I5

1.1 SOFTWARE REQUIREMENTS:

Rational rose /Argo UML

1.2 PROBLEM ANALYSIS AND PROJECT PLAN

To simplify the process of applying passport, software has been created by designing through rational rose tool. Initially the applicant login the passport automation system and submits his details. These details are stored in the database and verification process done by the passport administrator, regional administrator and police the passport is issued to the applicant.

1.3 PROJECT DESCRIPTION:

This software is designed for the verification of the details of the student by the central computer. The details regarding the student will be provided to the central computer through the administrator in the book bank and the computer will verify the details of student and provide approval to the office. Then the books that are needed by the student will issue from the office to the him.

1.4 REFERENCES:

IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

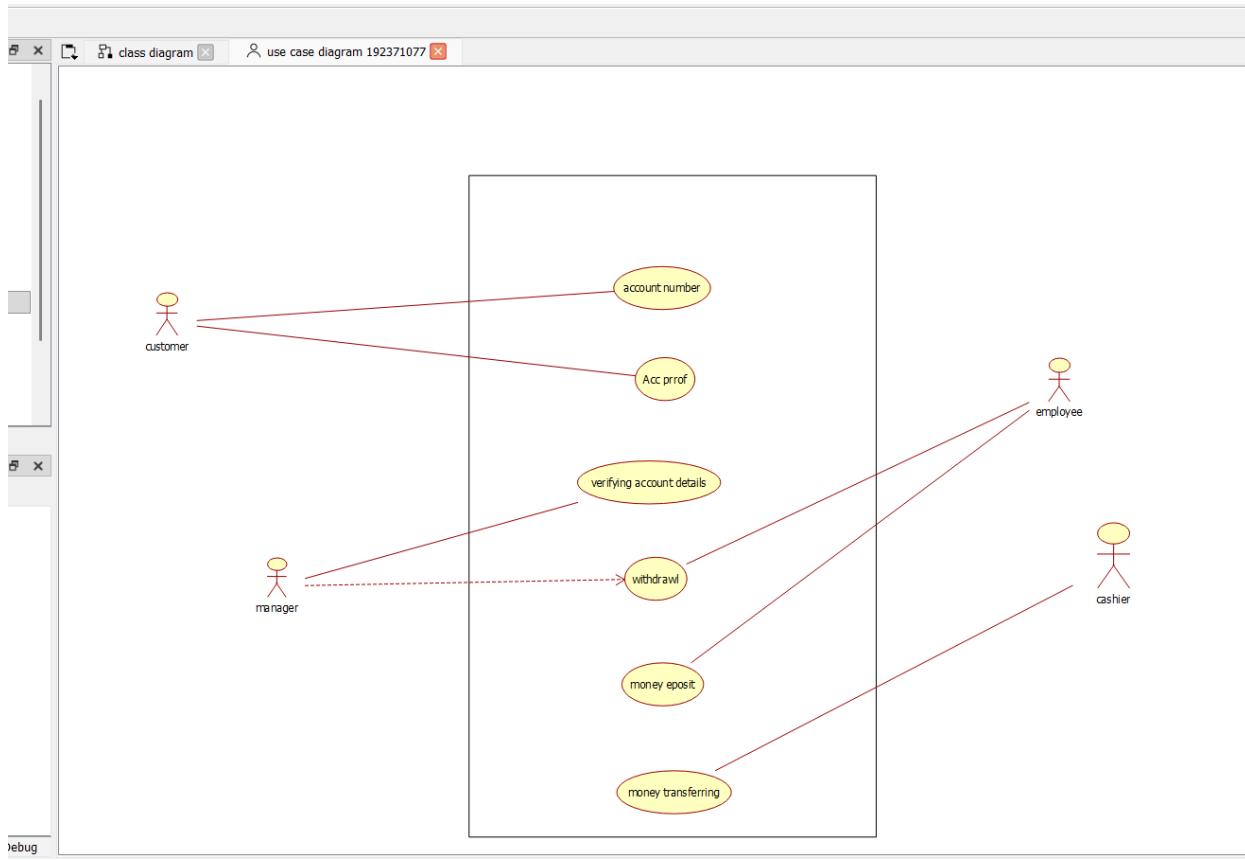
This diagram will contain the actors, use cases which are given below Actors:

Student, book bank admin.

REGISTER NO:192371077

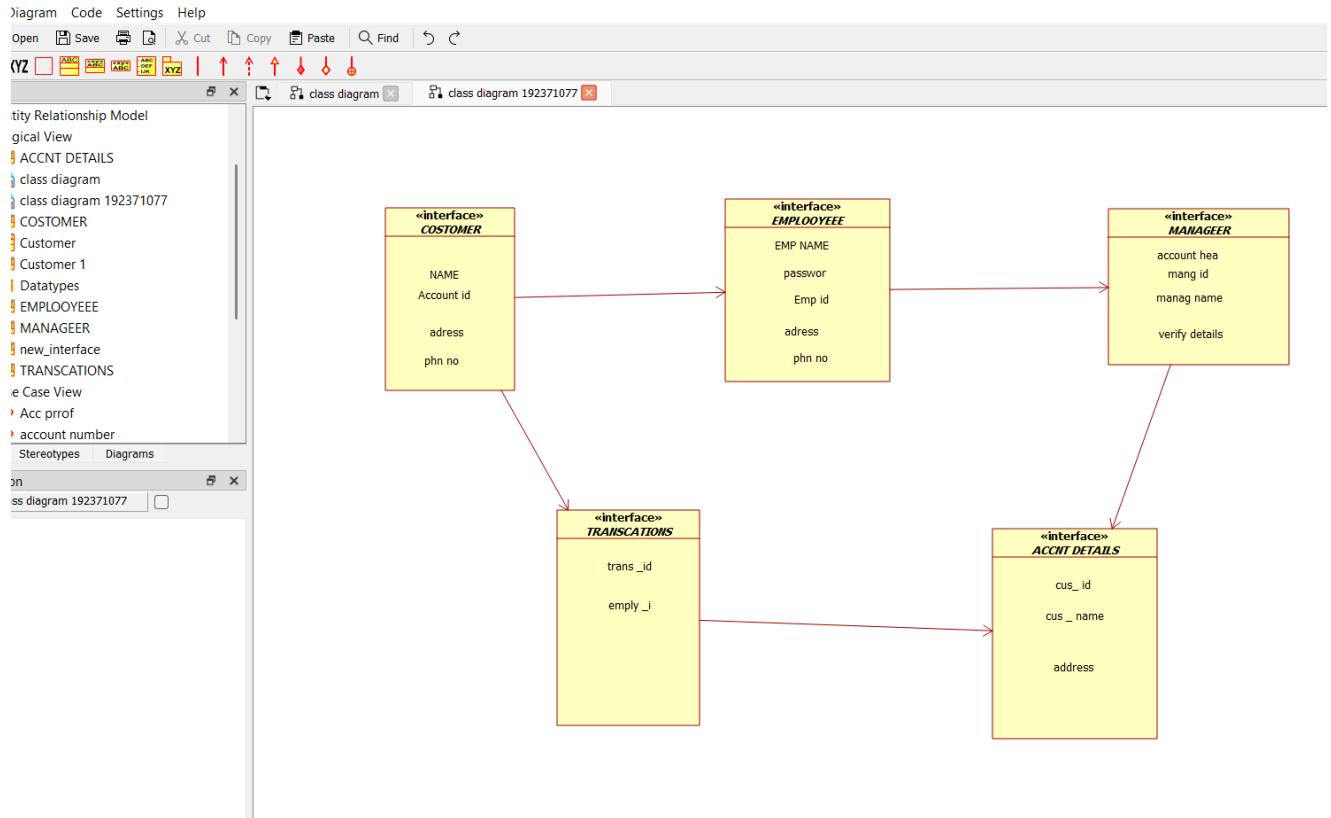
OOAD LAB

Use case: Student details, register, verify student id, return previous books, request of books, issue of books, check of book availability.



CLASS DIAGRAM:

This diagram consists of the following classes, attributes and their operations. Check availability()



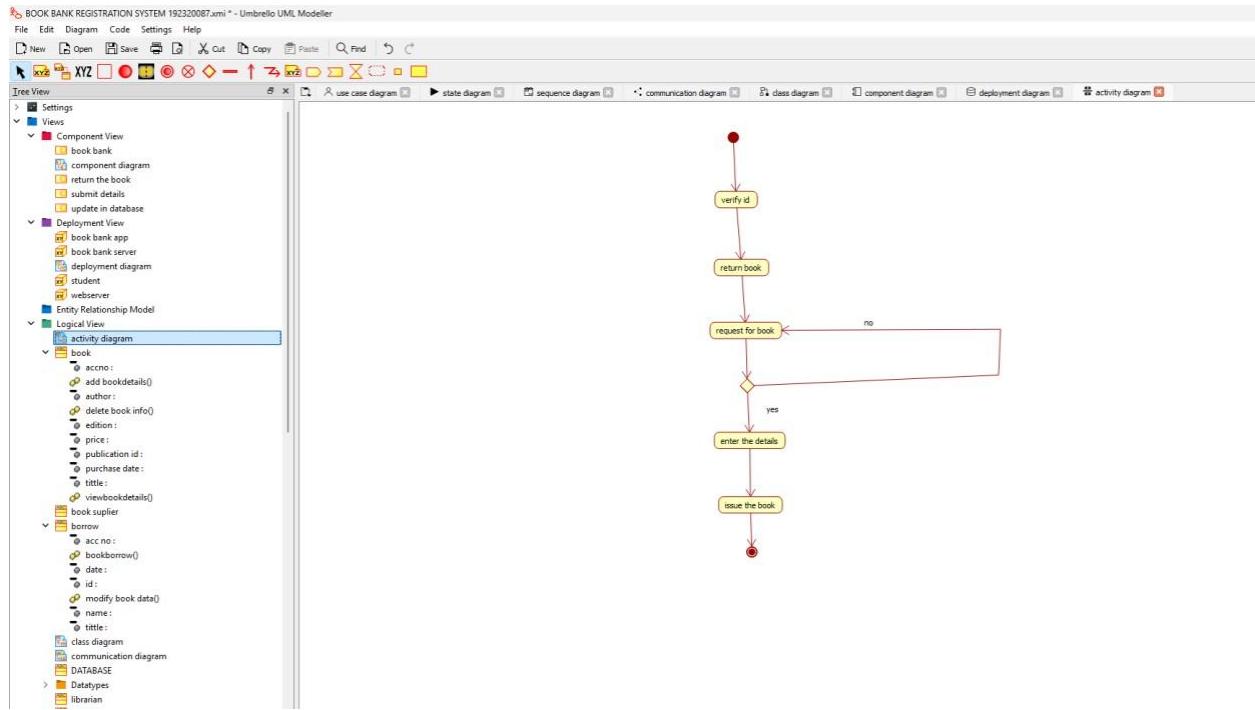
ACTIVITY DIAGRAM:

This diagram will have the activities as Start point, End point, Decision boxes as given below:

CLASSES	ATTRIBUTES	OPERATIONS
Computer	Student record, book list	Enter issue(), check availability()
Student	Student details	Request for book(), register()
admin	Student details, book list	Verify student id(), issue book()

Activities: Verify id, return books, request for books, enter book issue details in system, issue books

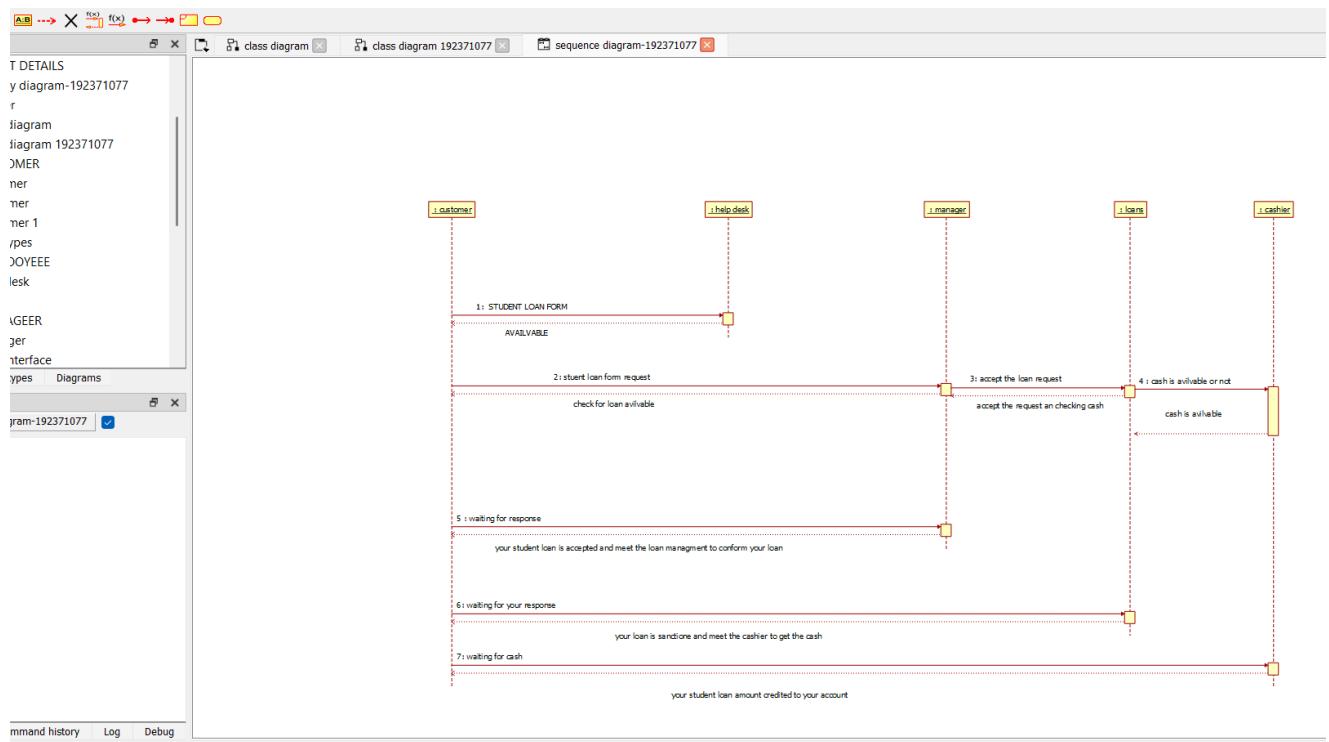
Decision box: Check availability of books whether it is present or not.



SEQUENCE DIAGRAM:

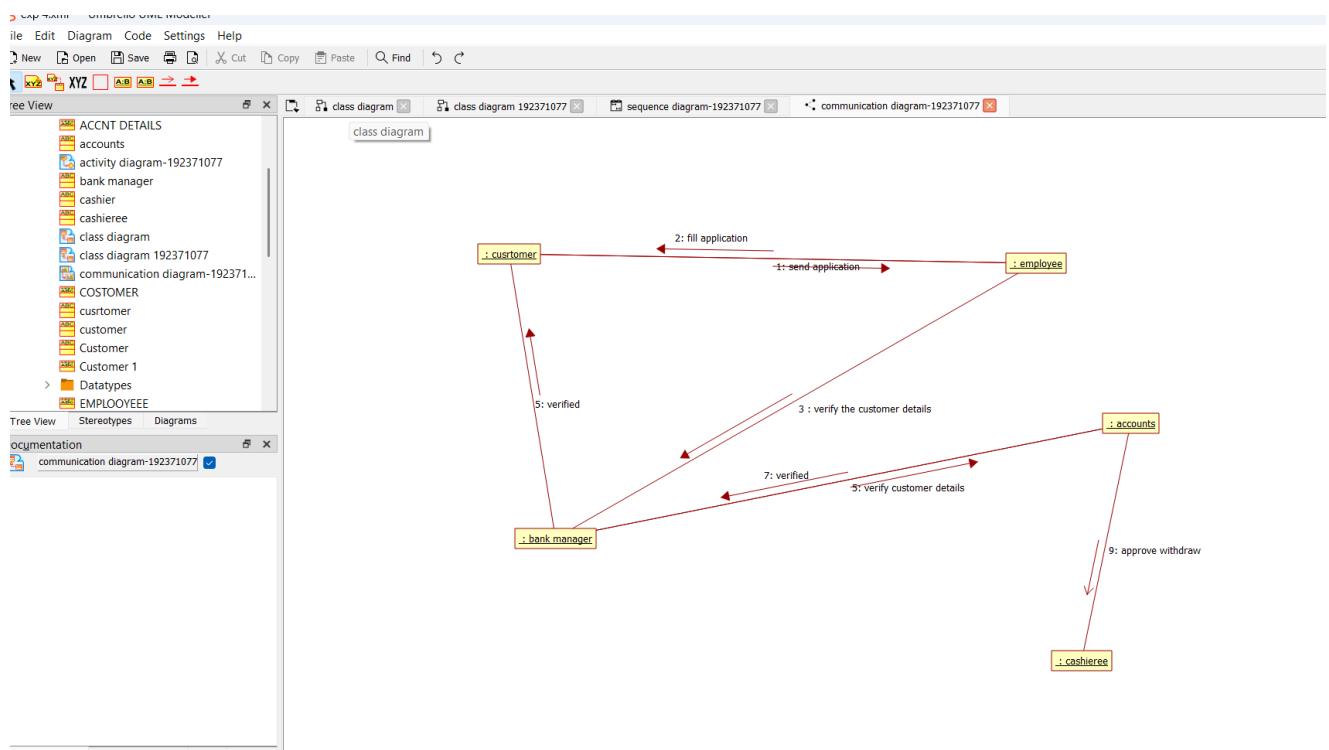
This diagram consists of the objects, messages and return messages.

Object: Stud, admin, computer



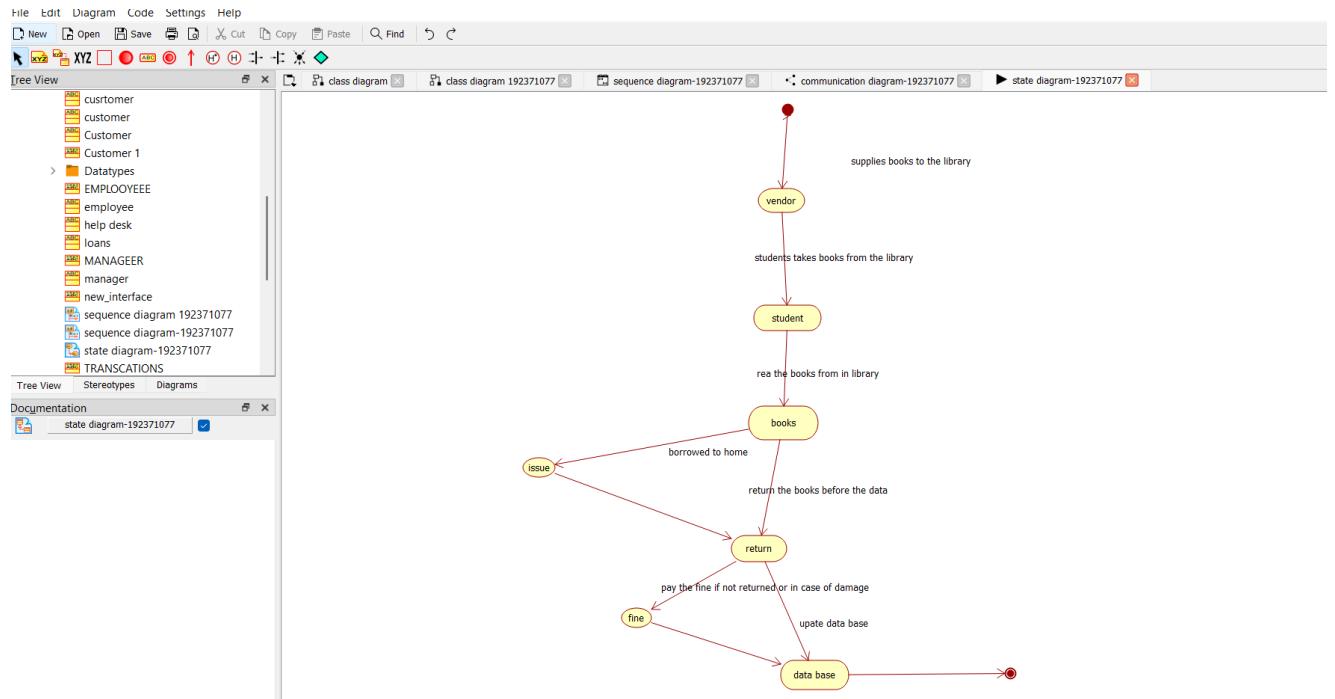
COLLABORATION DIAGRAM:

This diagram contains the objects and actors. This will be obtained by the completion of the sequence diagram and pressing the F5 key.



STATE CHART DIAGRAM

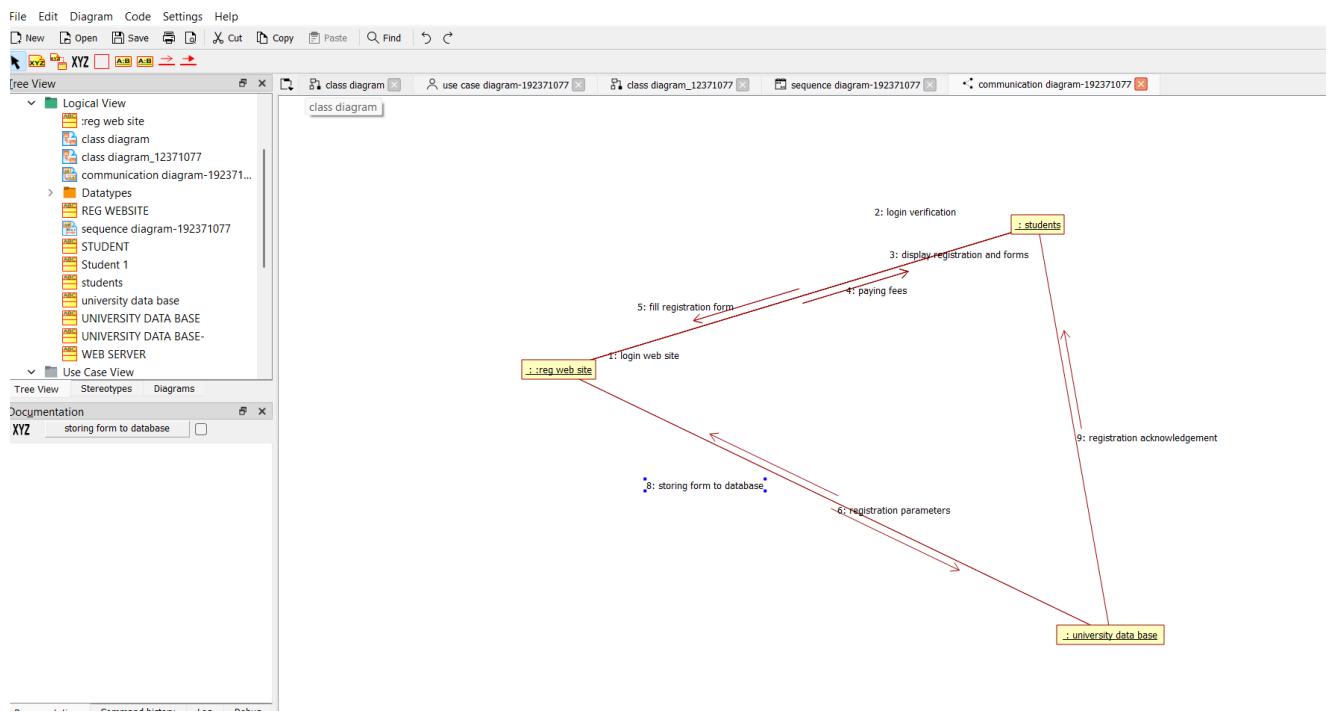
The purpose of state chart diagram is to understand the algorithm involved in performing a method. It is also called as state diagram. A state is represented as a round box, which may contain one or more compartments. An initial state is represented as small dot. An final state is represented as circle surrounding a small dot.



COMPONENT DIAGRAM

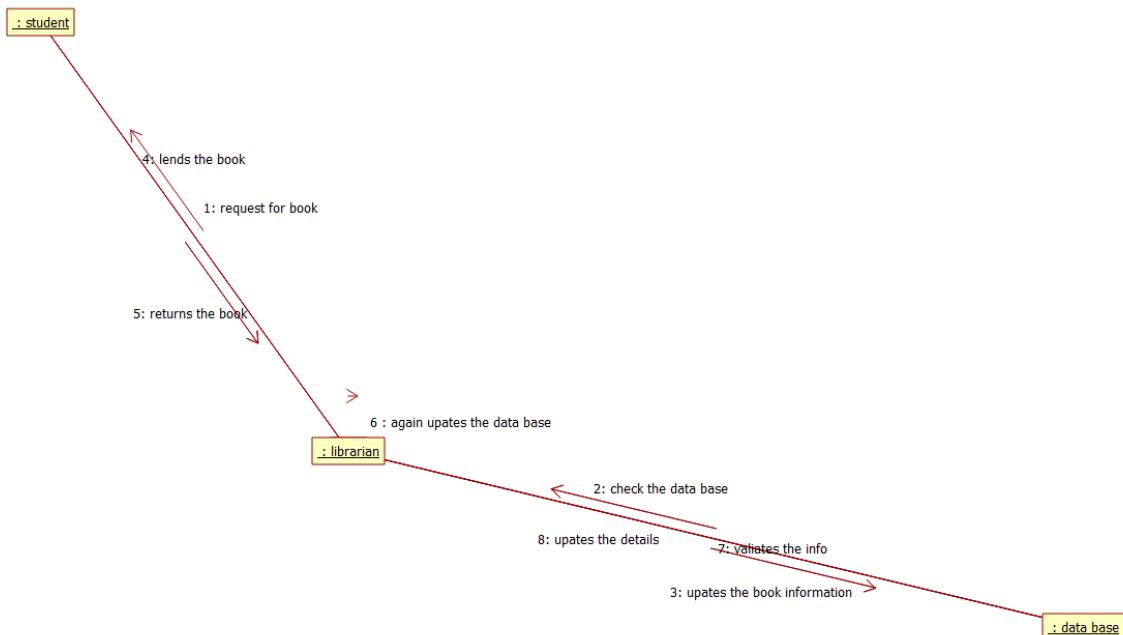
The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by boxed figure. Dependencies are represented by communication association

REGISTER NO:192371077



OOD LAB

DEPLOYMENT DIAGRAM



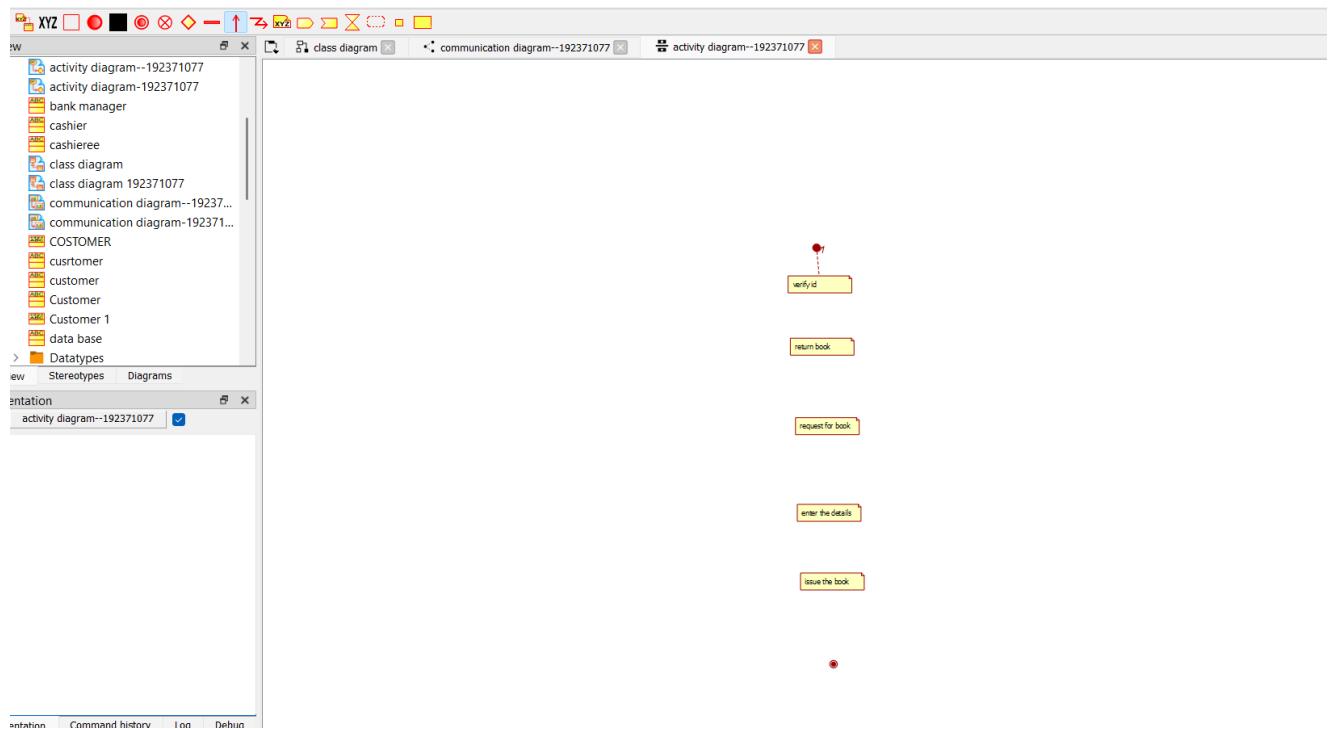
A deployment diagram in the unified modeling language serves to model the physical deployment of artifacts on deployment targets. Deployment diagrams show "the allocation of artifacts to nodes according to the Deployments defined between them. It is represented by 3-dimensional box. Dependencies are represented by communication association.

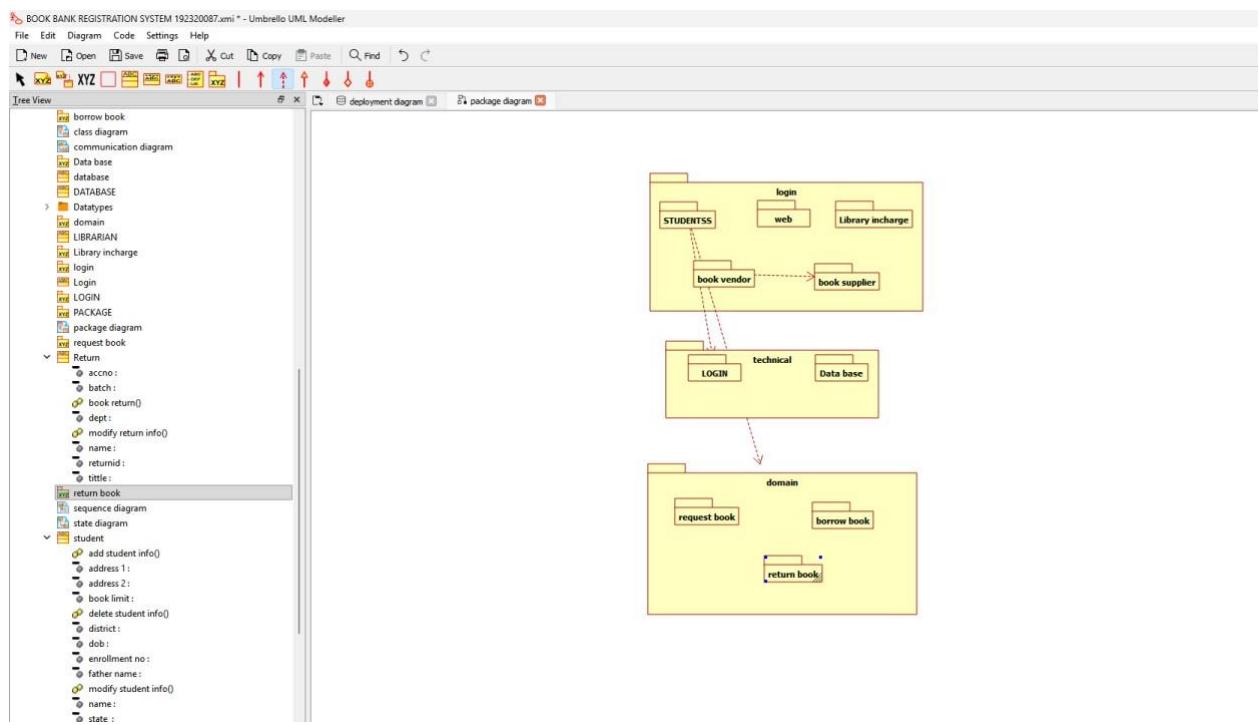
PACKAGE DIAGRAM:

A package diagram in unified modeling language that depicts the dependencies between the packages that make up a model. A Package Diagram (PD) shows a grouping of elements in the OO model, and is a Cradle extension to UML. PDs can be used to show groups of classes in Class Diagrams (CDs), groups of components or processes in Component Diagrams (CPDs), or groups of processors in Deployment Diagrams (DPDs).

There are three types of layer. They are

- o User interface layer o
- Domain layer o Technical services layer





PROGRAM CODING:

ADMIN:

Public class admin

{

 Public integer student details;

 Public integer book list;

 Public void verify stud id()

{

}

 Public void check for availability()

{

}

 Public void issue books()

```
{  
}
```

```
Public void order for new author()
```

REGISTER NO:

OOD LAB

```
{  
}
```

```
Public void maintains stud details()
```

```
{  
}  
}
```

STUDENT:

```
Public class stud
```

```
{
```

```
Public integer studdetails;
```

```
Public void request for books()
```

```
{  
}
```

```
Public void register()
```

```
{  
}
```

COMPUTER:

```
Public class computer
```

```
{
```

```
Public integer stud record;  
Public integer booklist;  
Public void maintain stud rec()  
{  
}  
Public void enter issue()  
{  
}
```

REGISTER NO:

OOAD LAB

```
Public void order new author()  
{  
}  
Public void check availability()  
{  
}  
}
```

RESULT:

Thus the diagrams [use case, activity, sequence, collaboration, class, collaboration, component, deployment, package] for the Book bank registration system has been designed, executed and output is verified.