EX.NO:09 DATE:	E-BOOK MANAGEMENT SYSTEM

AIM:

To draw the diagrams [usecase, activity, sequence, collaboration, class, statechart, component, deployment, package] for E-book management 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 PLANNING

E-book Management System gives an idea about how books are maintained in the particular websites. The books that are to be purchased, the books that are to be sold are maintained here. . Further some additional details of the current books that is available in the store are also given. E book Management System in this project is done in an authorized way. The password and user id has been set here.

1.3 PROJECT DESCRIPTION:

This software is designed to manage the books that were read through the internet. This consists of the details of the e-book that were read by the user online. It will be controlled by the central system. This system act as a backup of all details together.

1.4 REFERENCES:

IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

This diagram will contain the actors, use cases which are given below
Actors: user, e-book management
Use case: login ,search books, download ,pay for the books, logout .

CLASS DIAGRAM:

This diagram consists of the following classes, attributes and their operations.

CLASSES	ATTRIBUTES	OPERATIONS
Internet	Enter id, login, logout	Surf book()
User	Login, logout	Surf book()
E-book management system	verify user	check availability()

ACTIVITY DIAGRAM:

This diagram will have the activities as Start point ,End point, Decision boxes as given below:
Activities: Search for the e-book site,search for the book,download book **Decision box:** check availability

SEQUENCE DIAGRAM:

This diagram consists of the objects, messages

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.

STATECHART DIAGRAM:

It is a technique to describe the behavior of the system. It describes all the possible states that a particular object gets into the object oriented technique. State diagram are drawn for a single class to show to the lifetime behaviour of a single objects.

COMPONENT DIAGRAM:

The component diagram is represented by figure dependency and it is a graph of design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a systems. It is represented by boxed figure. Dependencies are represented by communication association.

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- dimentional 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:

E-BOOK MANAGEMENT:

```
public class e-book management extends internet
{

        public Integer enterid;
```

```
        public Integer verifyuser;

        public void checkavailability()
         {
         }

}
```

USER:
```
public class user1 extends internet
{
        public Integer login;
  public Integer logout;

        public void surfbooks()
         {
         }

}
```

CENTRAL SYSTEM:

```
public class central system
{
  public Integer enterid;
  public Integer download;
  public Integer login;
  public Integer logout;

         public void verify()
        {
         }

        public void status()
        {
         }

}
```

RESULT:

Thus the draw the diagrams [usecase, activity, sequence, collaboration, class, statechart, component, deployment, package] for E-book management system.

OOAD LAB

OOAD LAB