REGISTER NO:

Ex No:1	PASSPORT AUTOMATION SYSTEM
Date:	

AIM:

To draw the diagrams [usecase, activity, sequence, collaboration, class, statechart, collaboration, component, deployment, package] for the Passport Automation System.

SOFTWARE REQUIREMENTS SPECIFICATION

	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 passport details of the applicant by the central computer. The details regarding the passport will be provided to the central computer and the computer will verify the details of applicant and provide approval to the office. Then the passport will issue from the office to the applicant.

1.4 REFERENCES:

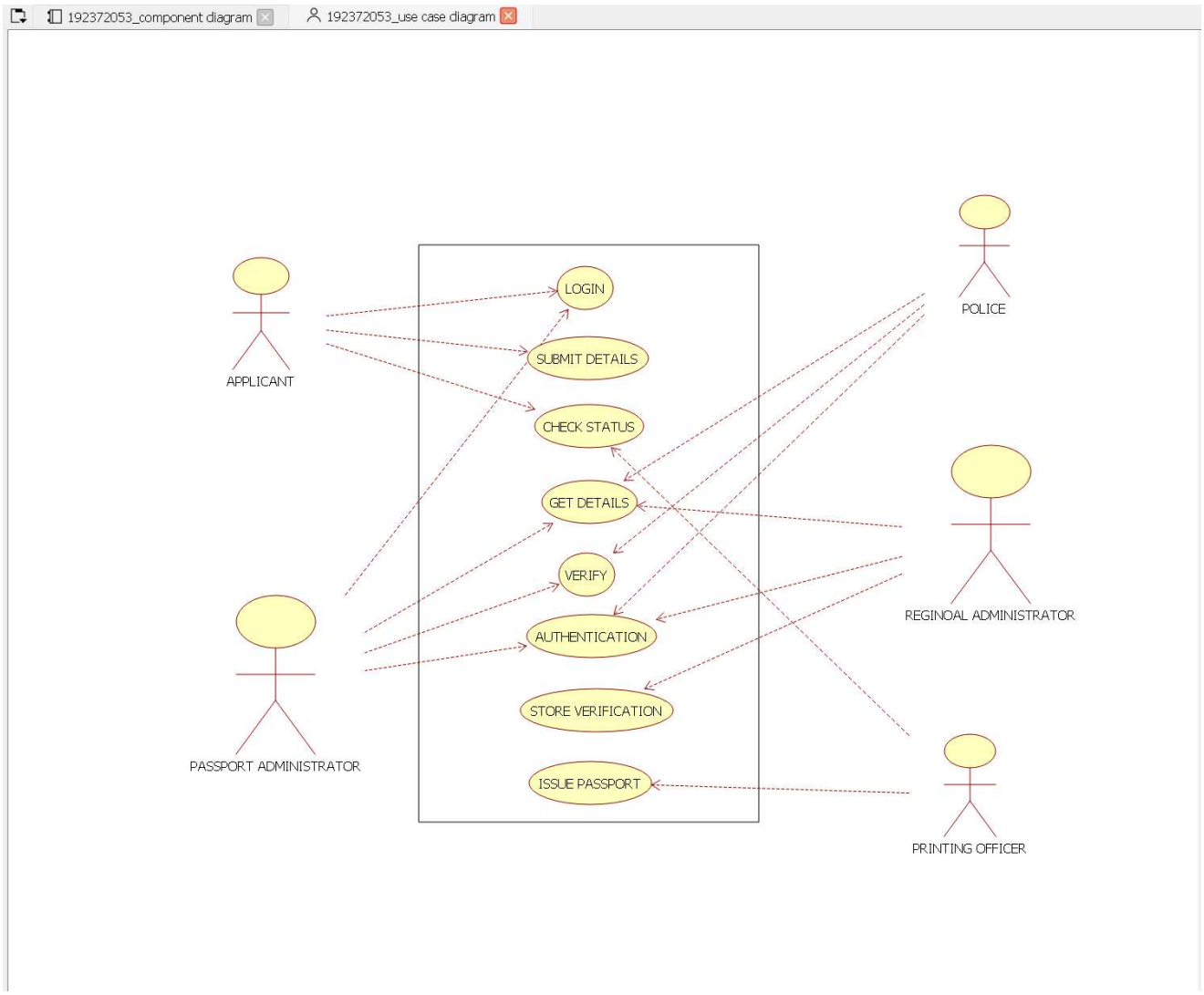
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Applicant, Enquiry Officer.

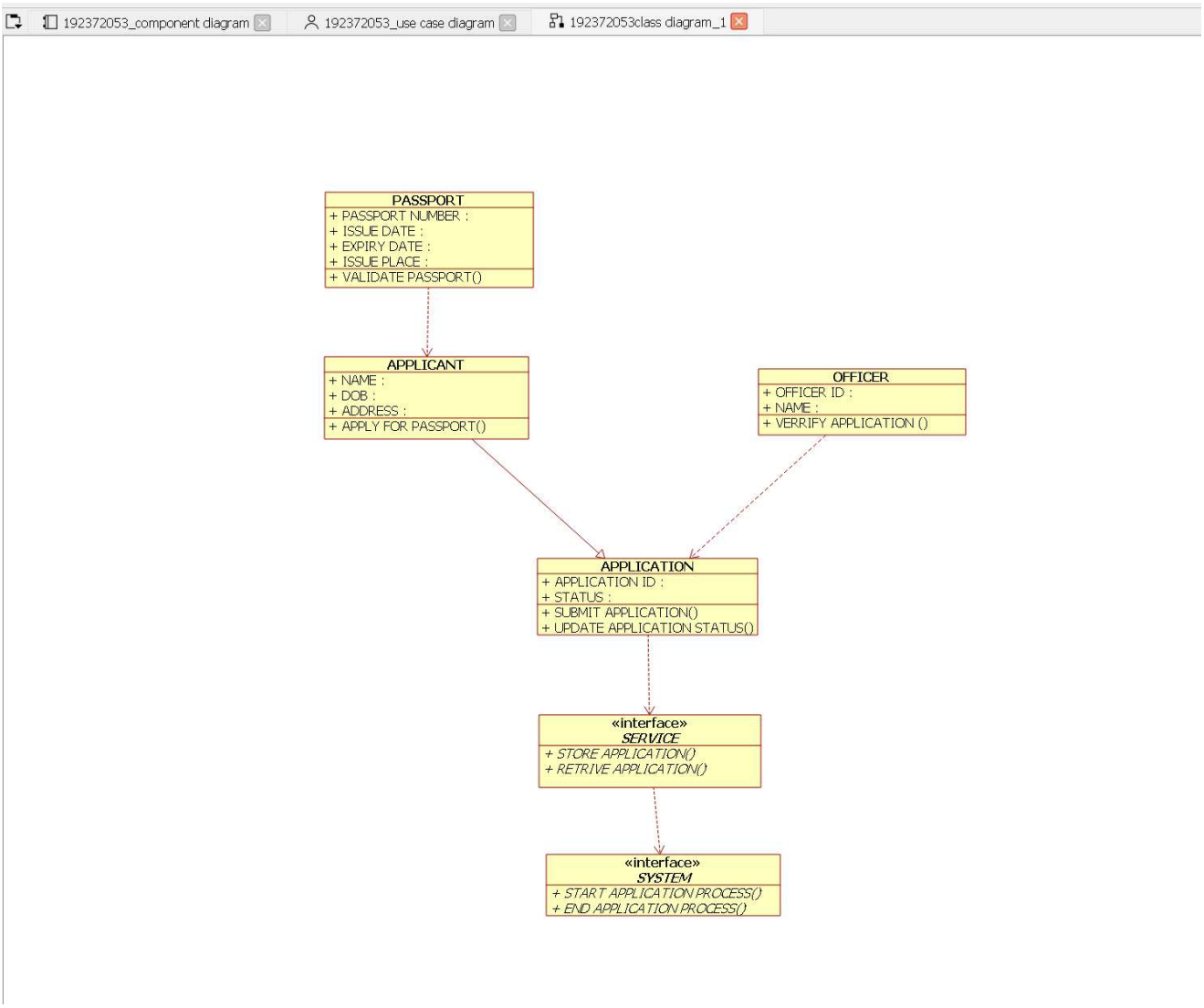
Use case: Applicant details, Applicant proof, Verification of proof, Issue of passport, Cancellation of the passport.



CLASS DIAGRAM:

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

CLASSES	ATTRIBTES	OPERATIONS
Passport management system	Verify details, Store proof	Verification of proof()
Enquiry officer	Applicant details	Issue of passport()
Applicant	Name, Details	Apply passport()

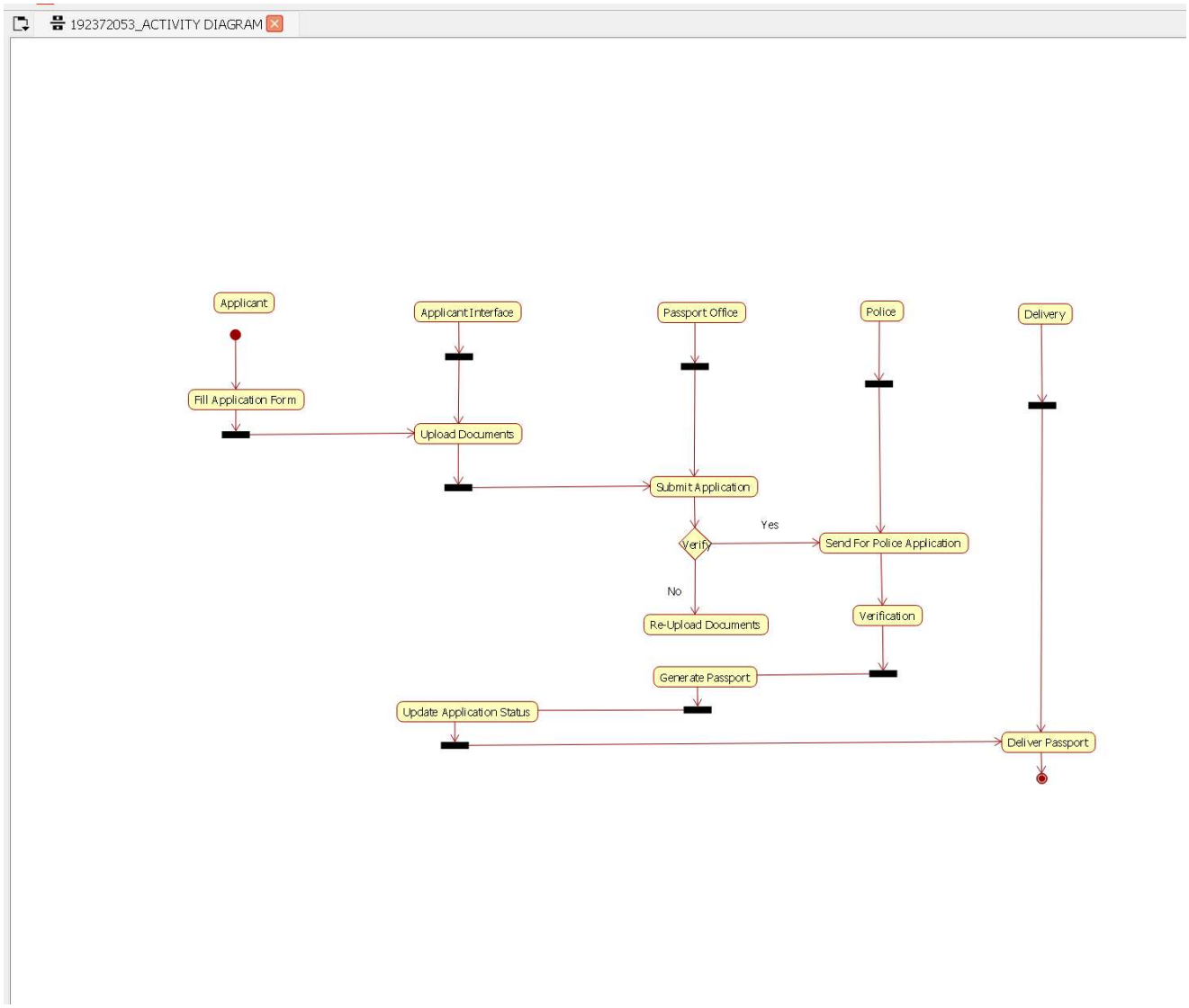


ACTIVITY DIAGRAM:

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

Activities: Enter applicant details, Submission of proof, Verification of details, Issue of passport.

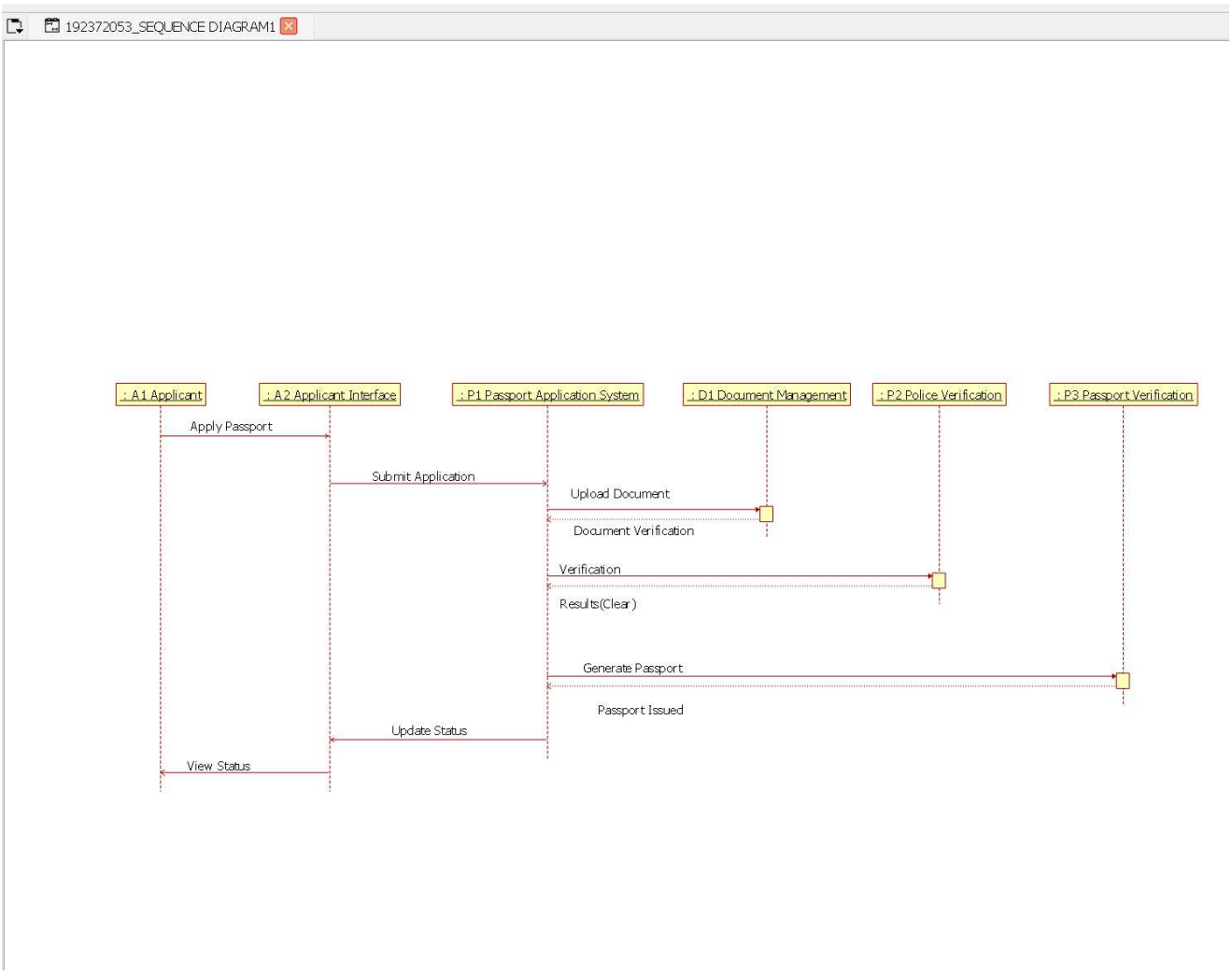
Decision box: Check details whether it is correct or not



SEQUENCE DIAGRAM:

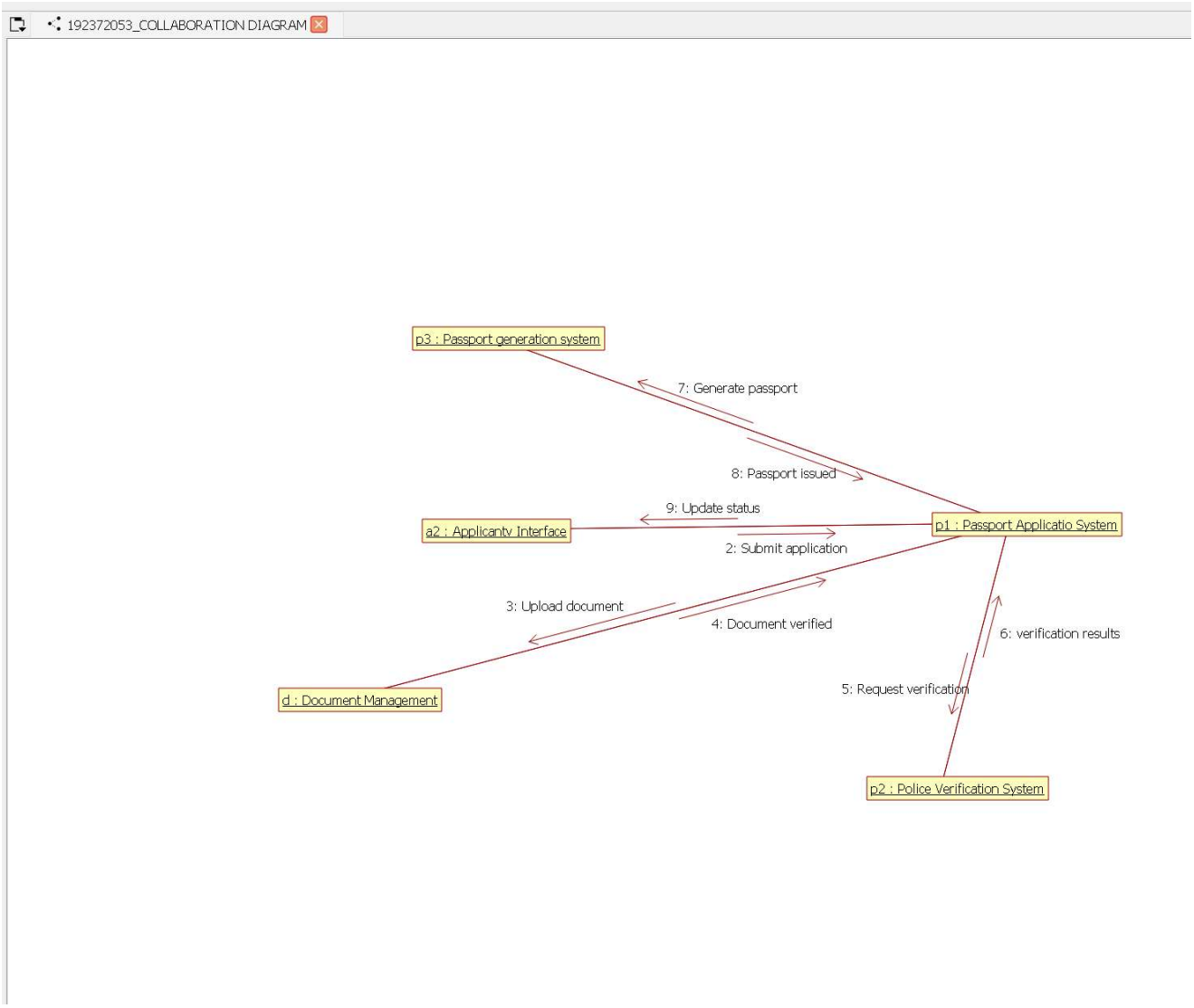
This diagram consists of the objects, messages and return messages.

Object: Applicant, Enquiry officer, Passport management system.



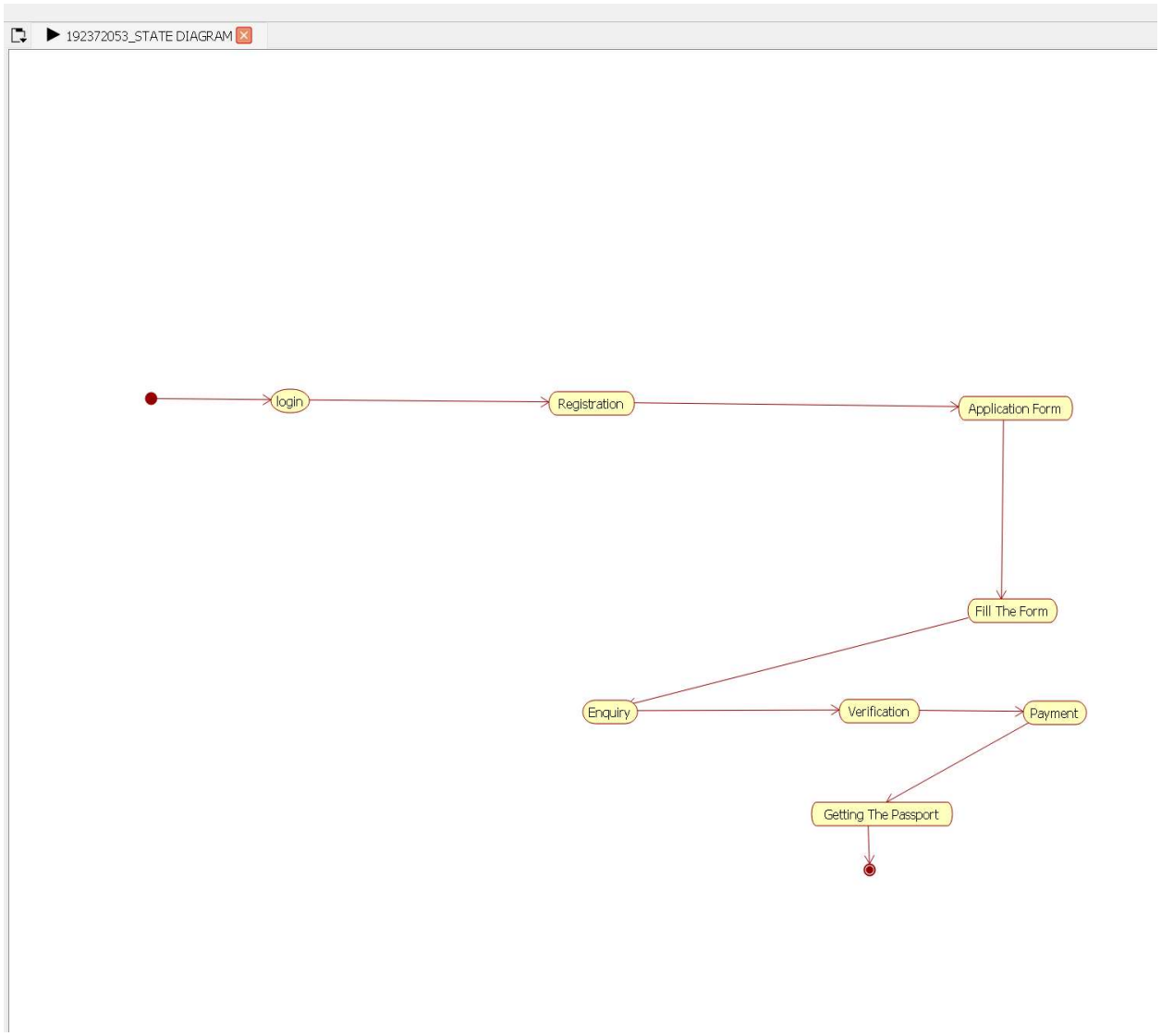
COLLABORATION DIAGRAM:

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



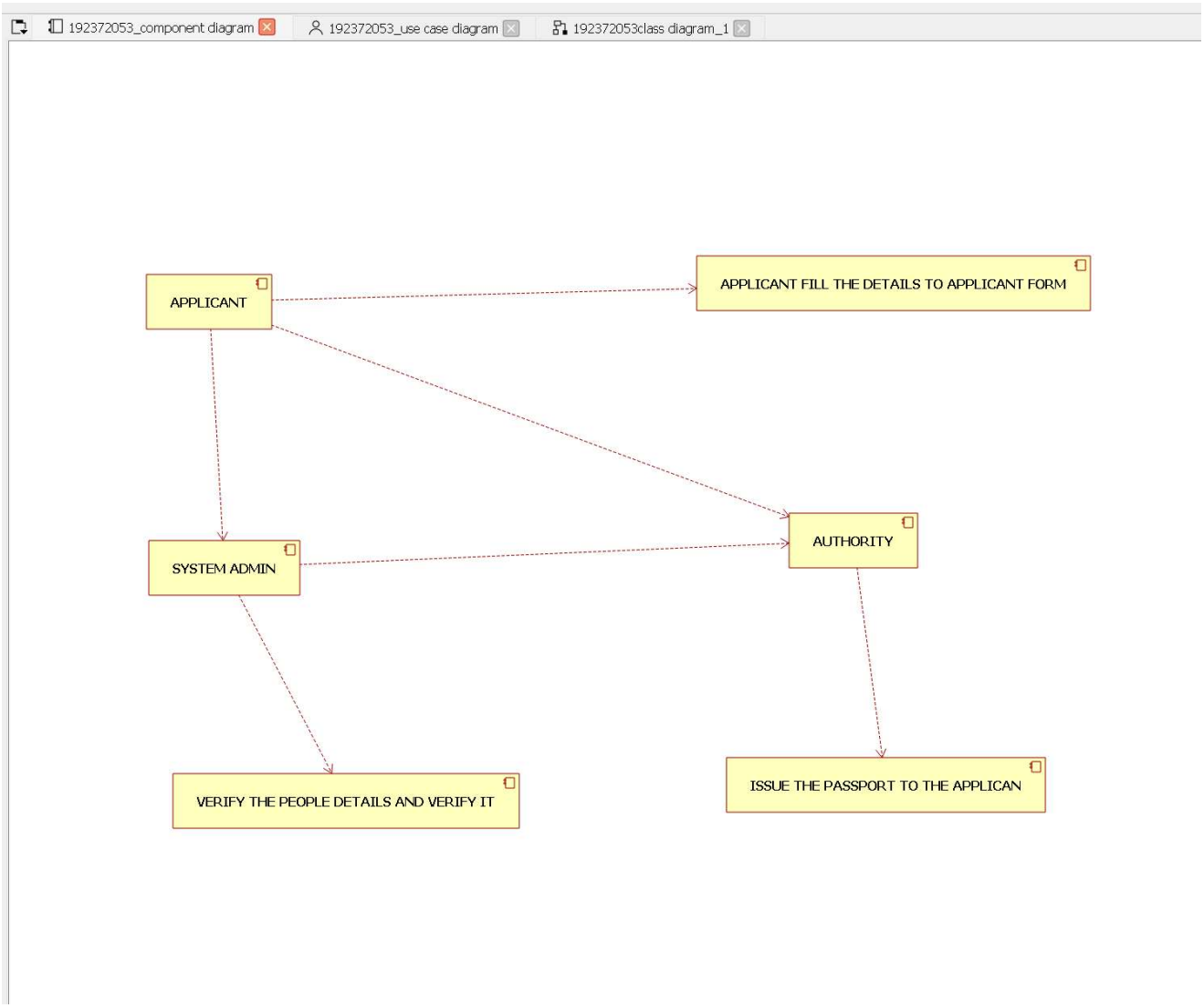
STATE CHART DIAGRAM :

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.



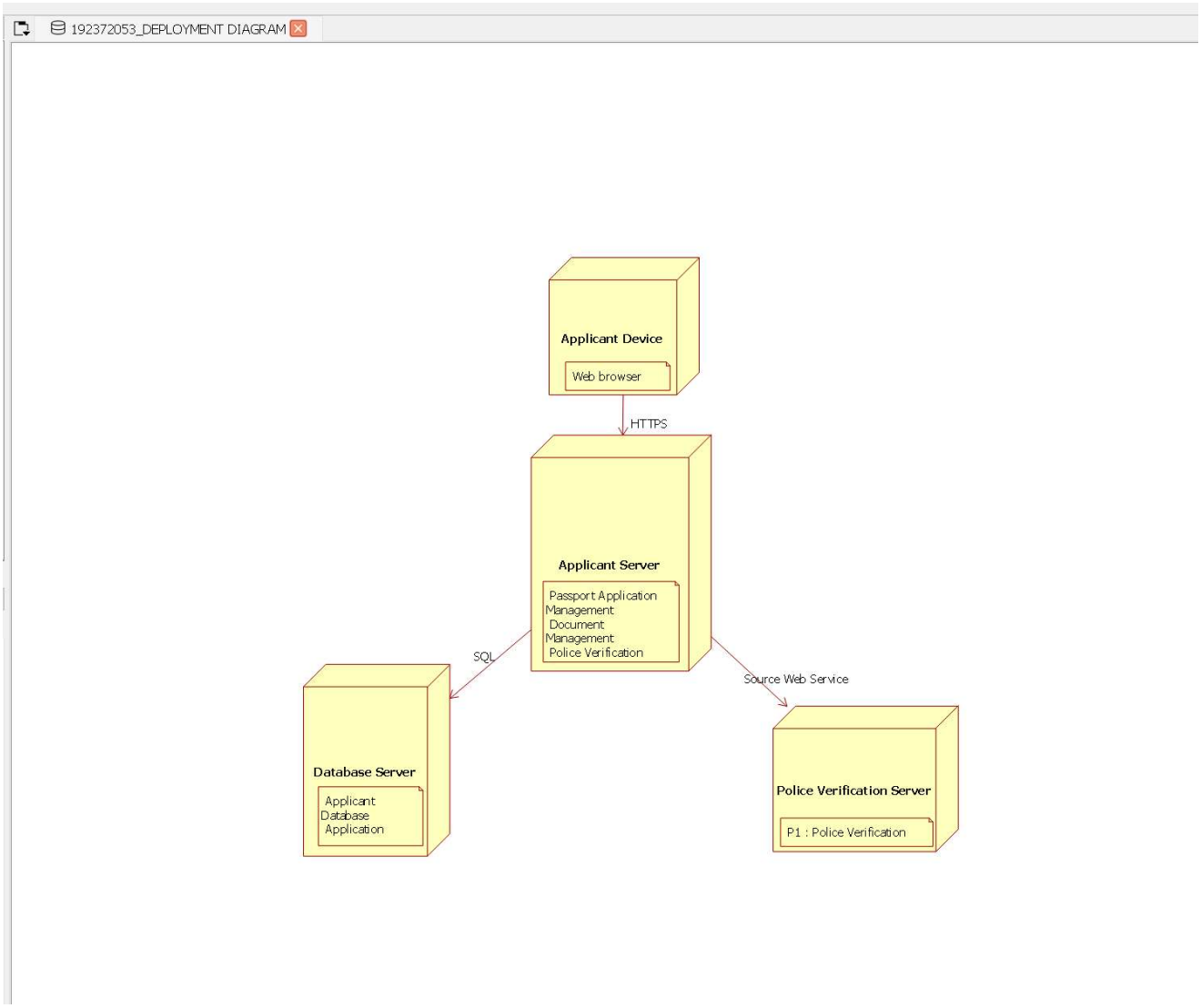
COMPONENT DIAGRAM

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.



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.



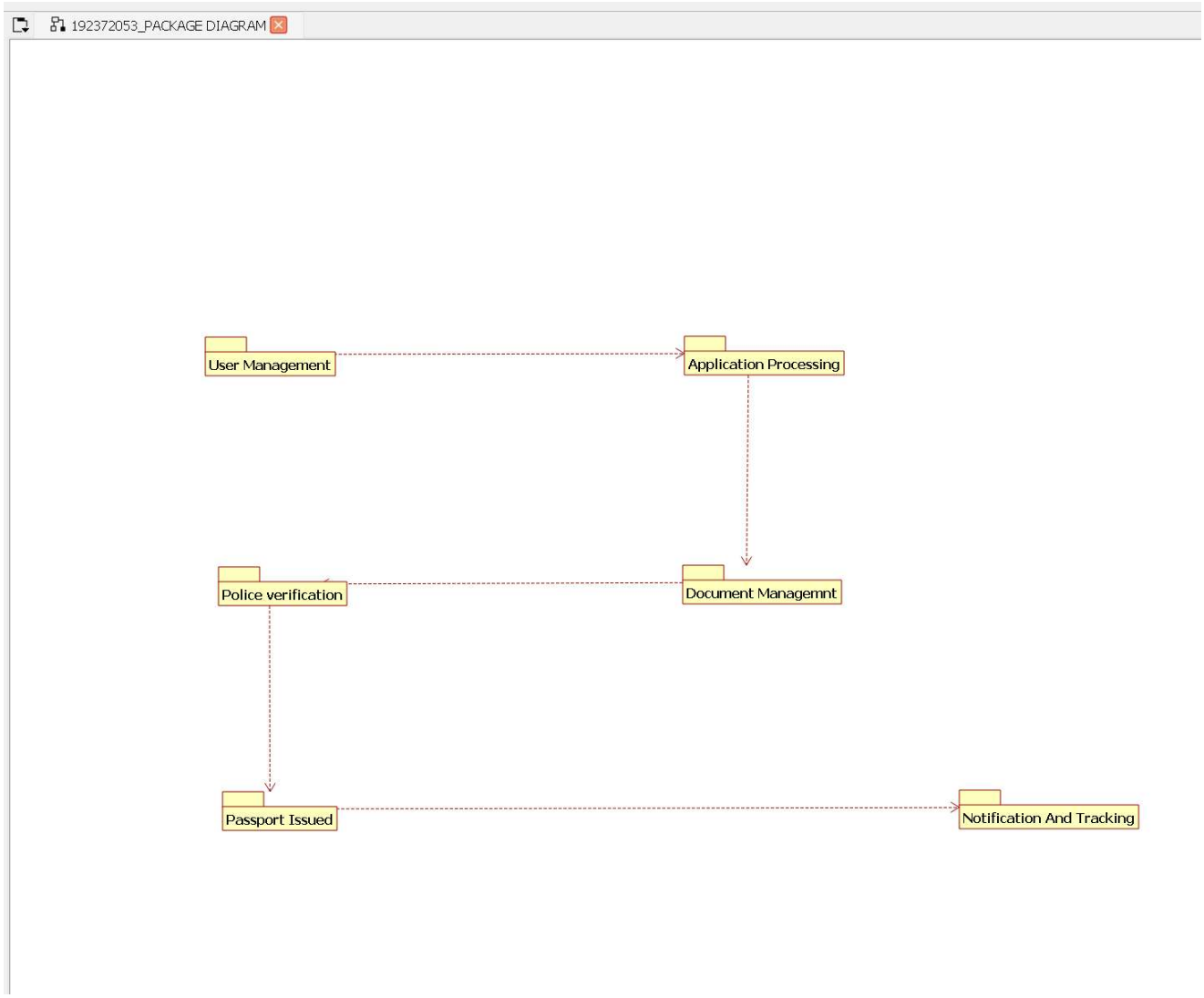
REGISTER NO:

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:

APPLICANT:

```
Public class Applicant
 {
        Public Integer firstname;
        Public Integer lastname;
        Public void passport()
         {
         }
}
```

PASSPORT APPLICATION SYSTEM:

```
Public class passport application system
 {
        Public Integer details;
        Public Integer proof;
        Public class Applicant
         {
        Public Integer firstname;
        Public Integer lastname;
```



```
        Public void passport()
         {
         }
        Public void verification()
         {
         }
        Public void issue()
         {
         }
        Public void cancel()
         {
         }
}
```

OFFICER:

```
Public class officer
 {
        Public Integer form;
```

```
        Public Integer responsible;
        Public void Database()
         {
         }
}
```

RESULT:

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

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

	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.

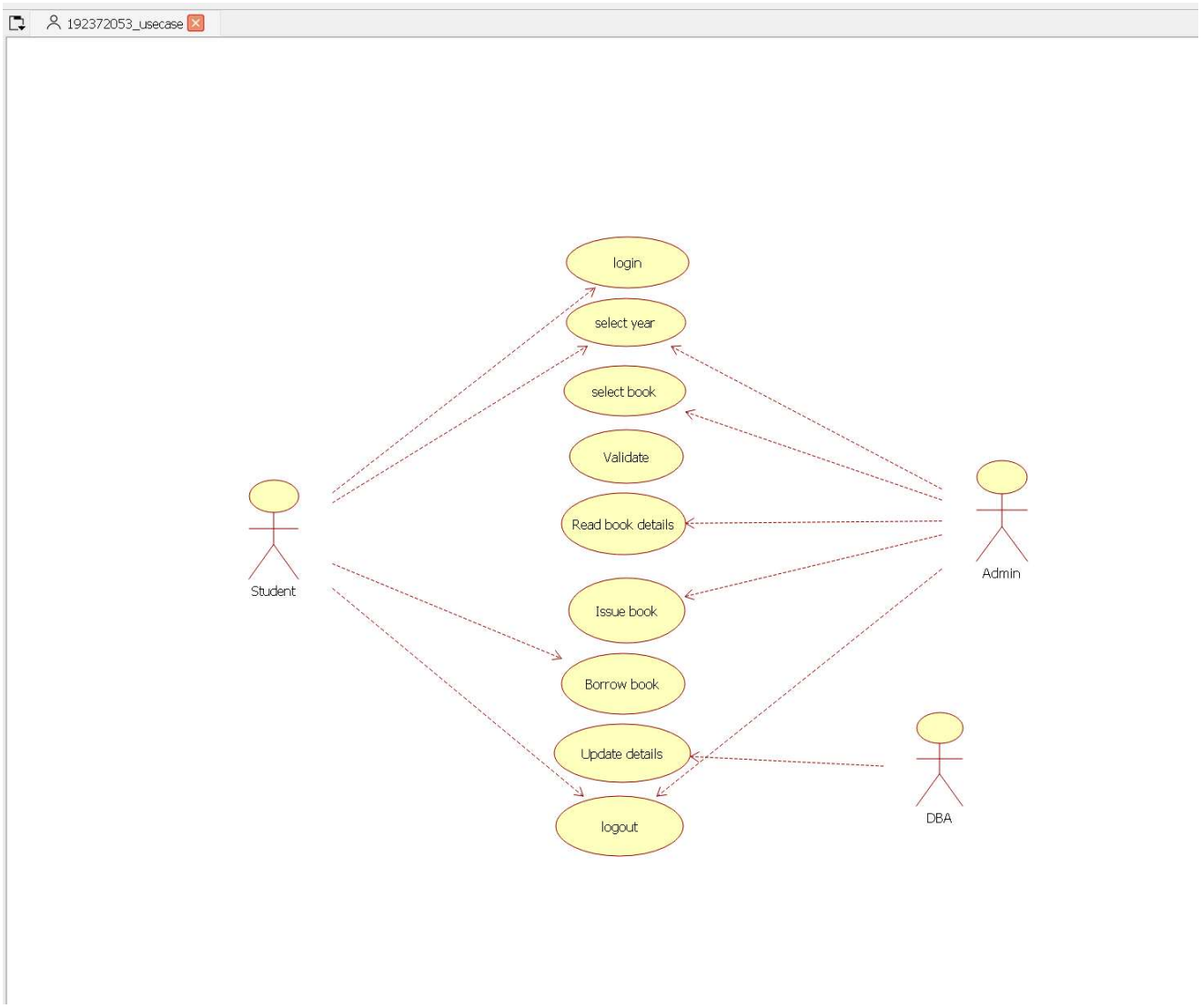
1.4 REFERENCES:

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.



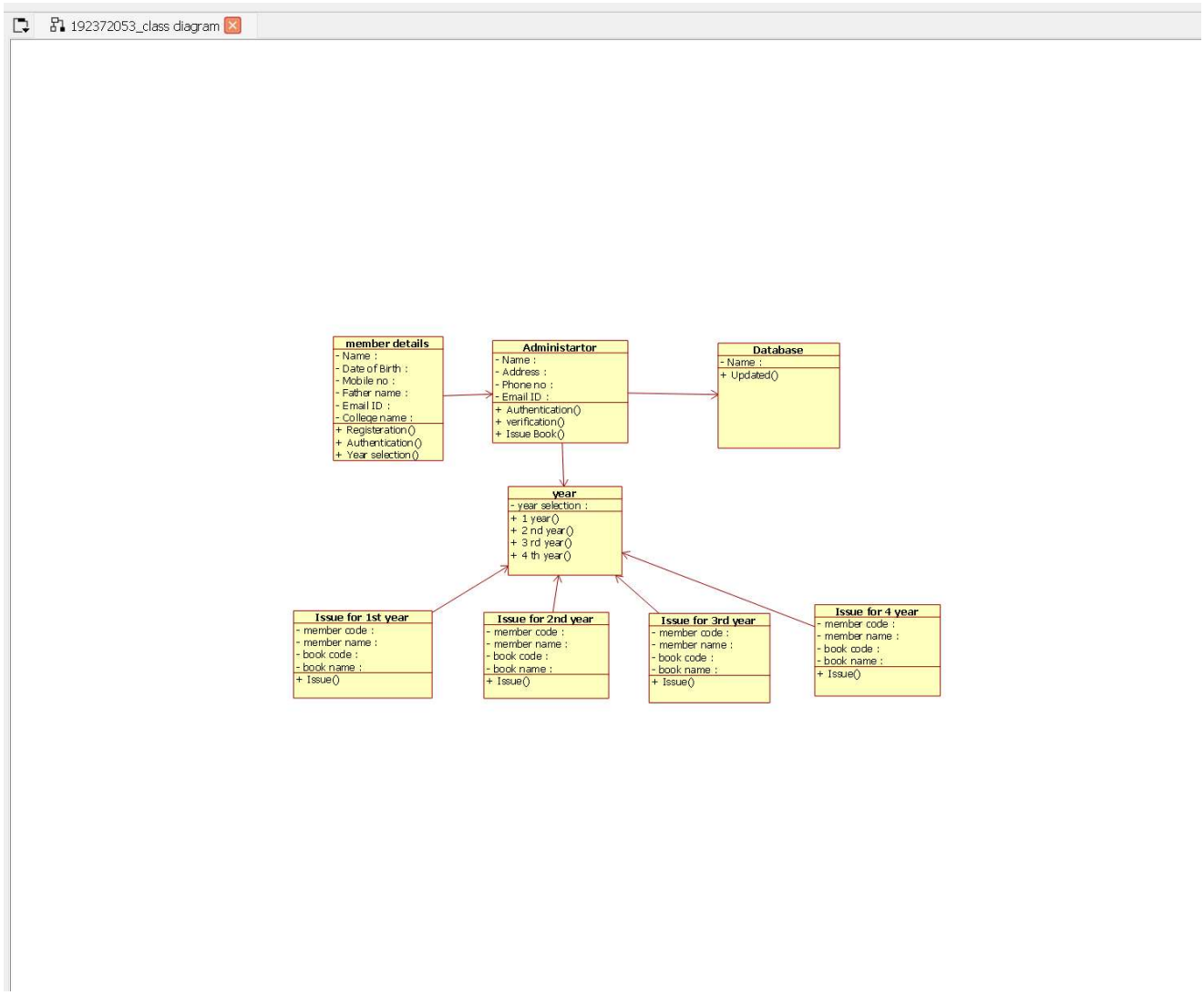
REGISTER NO:

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.

CLASSES	ATTRIBTES	OPERATIONS
Computer	Student record, Book list	Enter issue(), Check availability()
Stud	Student details	Request for books(), Register()
Admin	Student details, Book list	Verify student id(), Issue books()

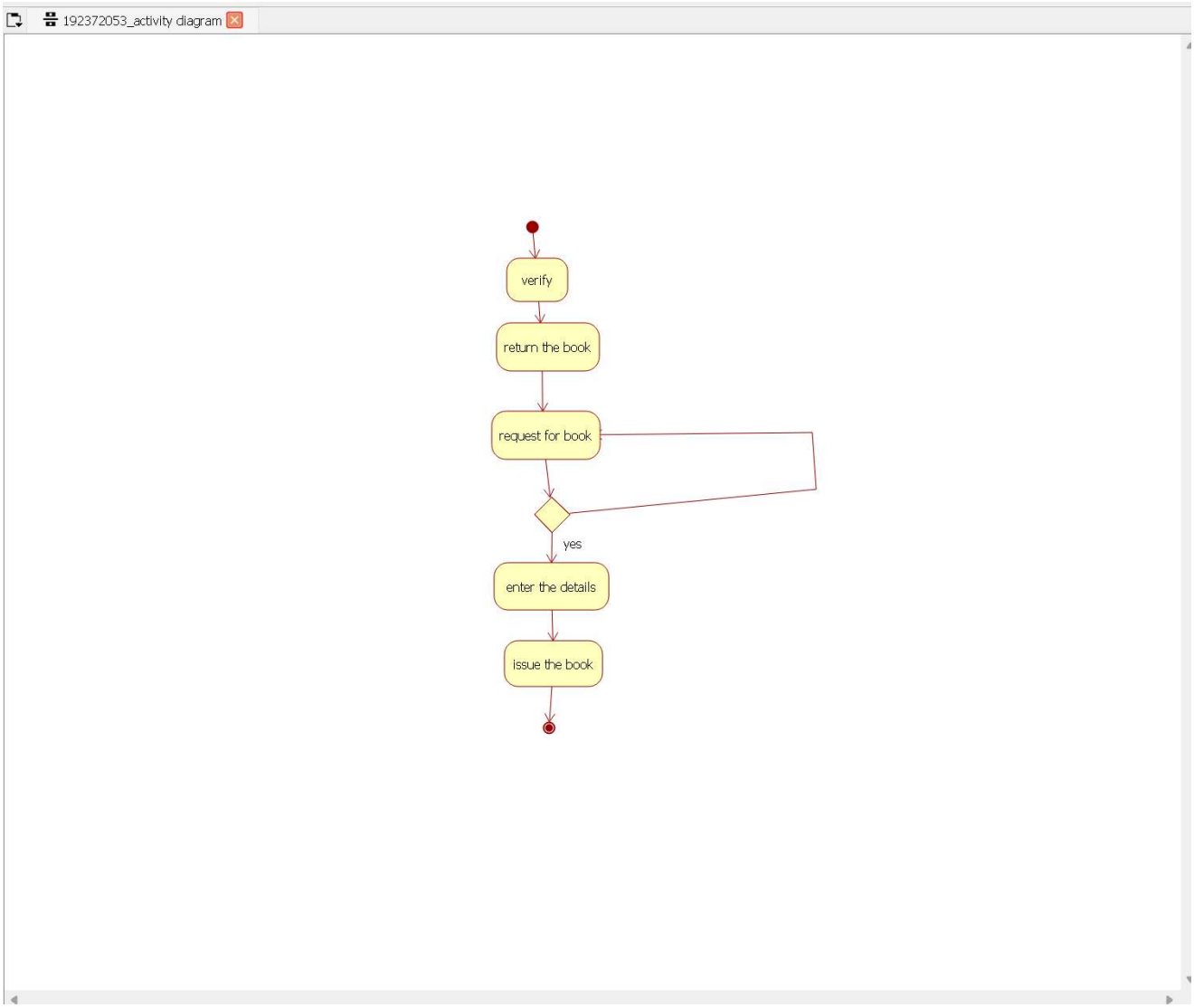


ACTIVITY DIAGRAM:

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

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

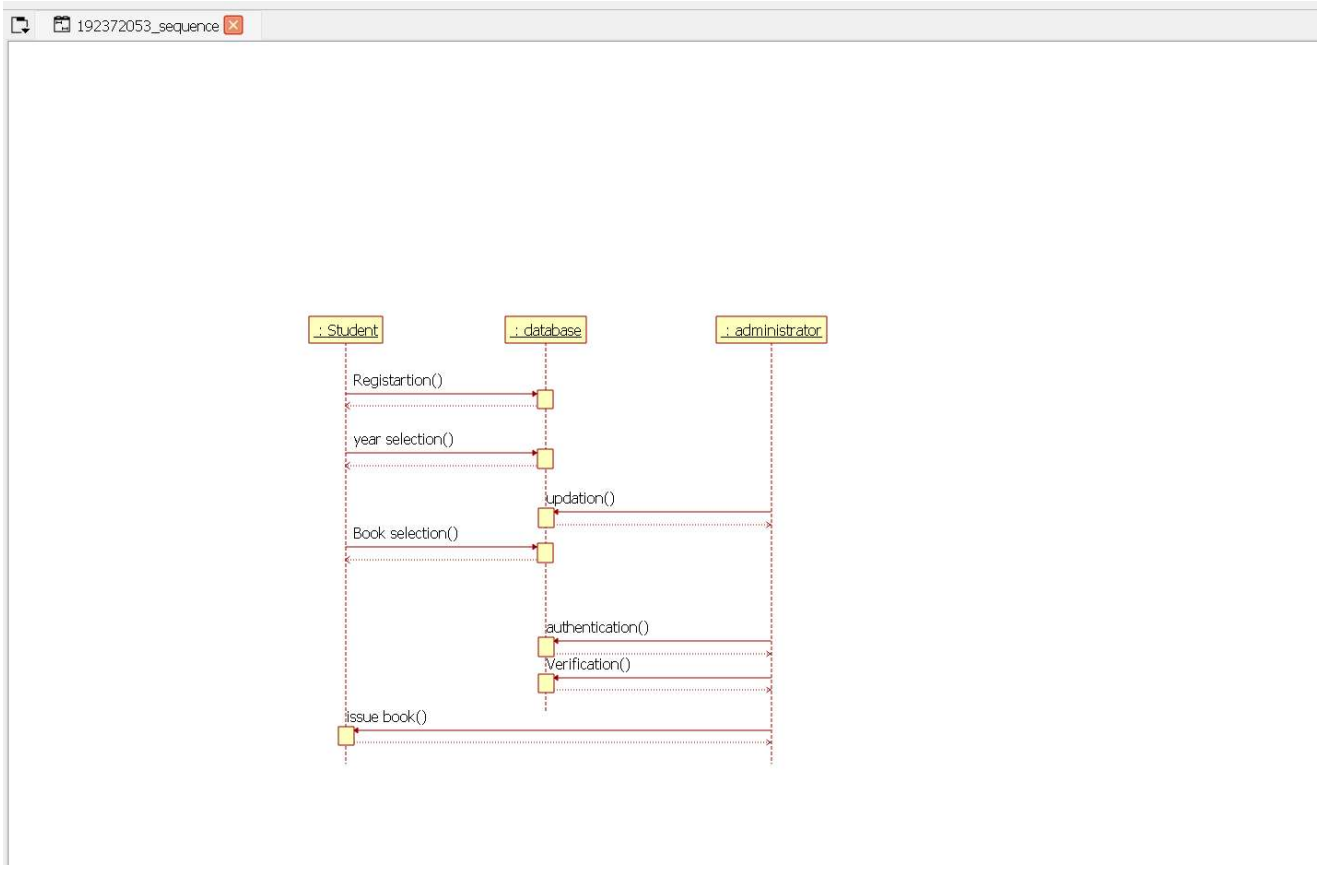
Decision box: Check availability of books whether it is present or not.



SEQUENCE DIAGRAM:

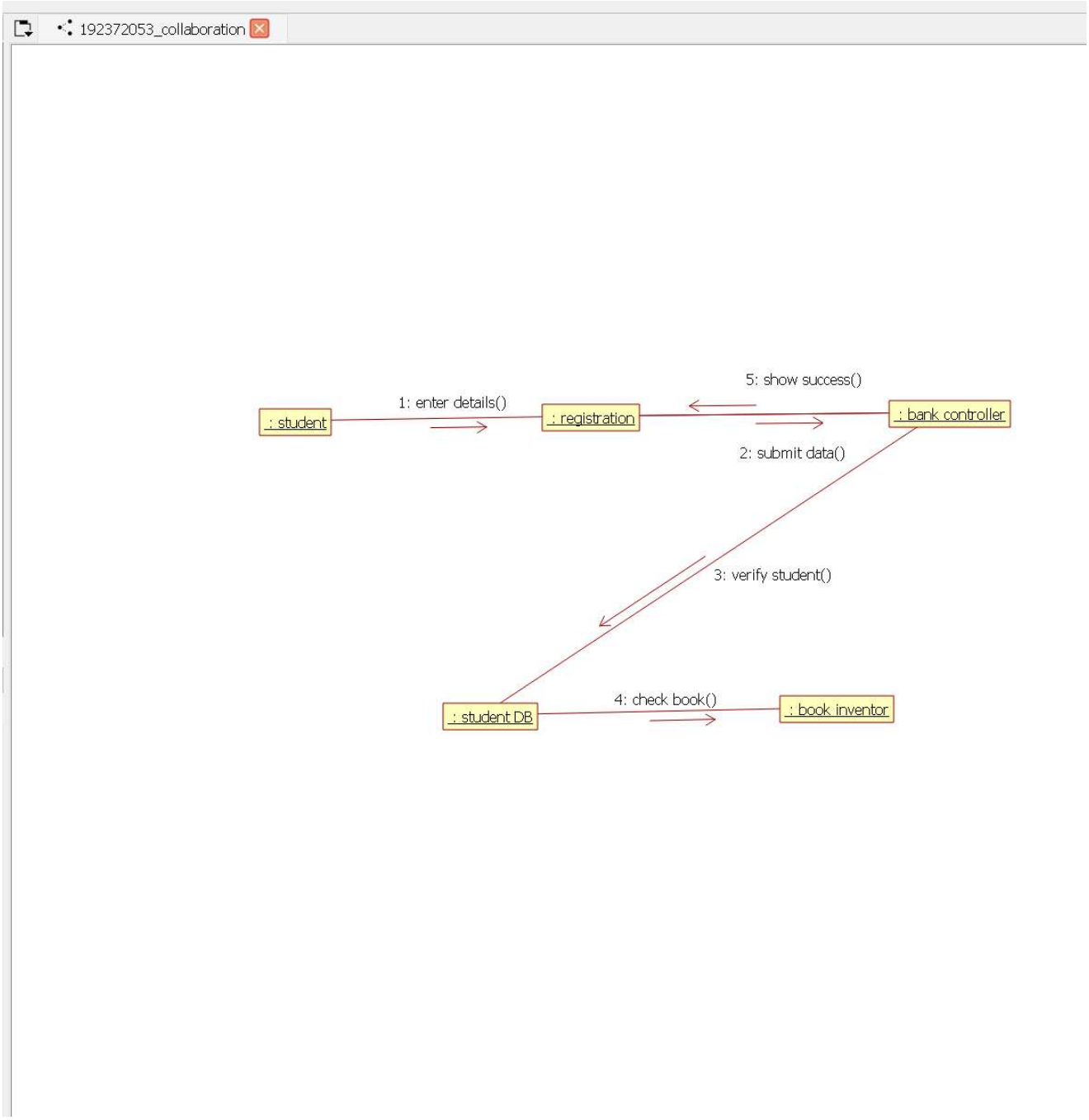
This diagram consists of the objects, messages and return messages.

Object: Stud, admin, computer



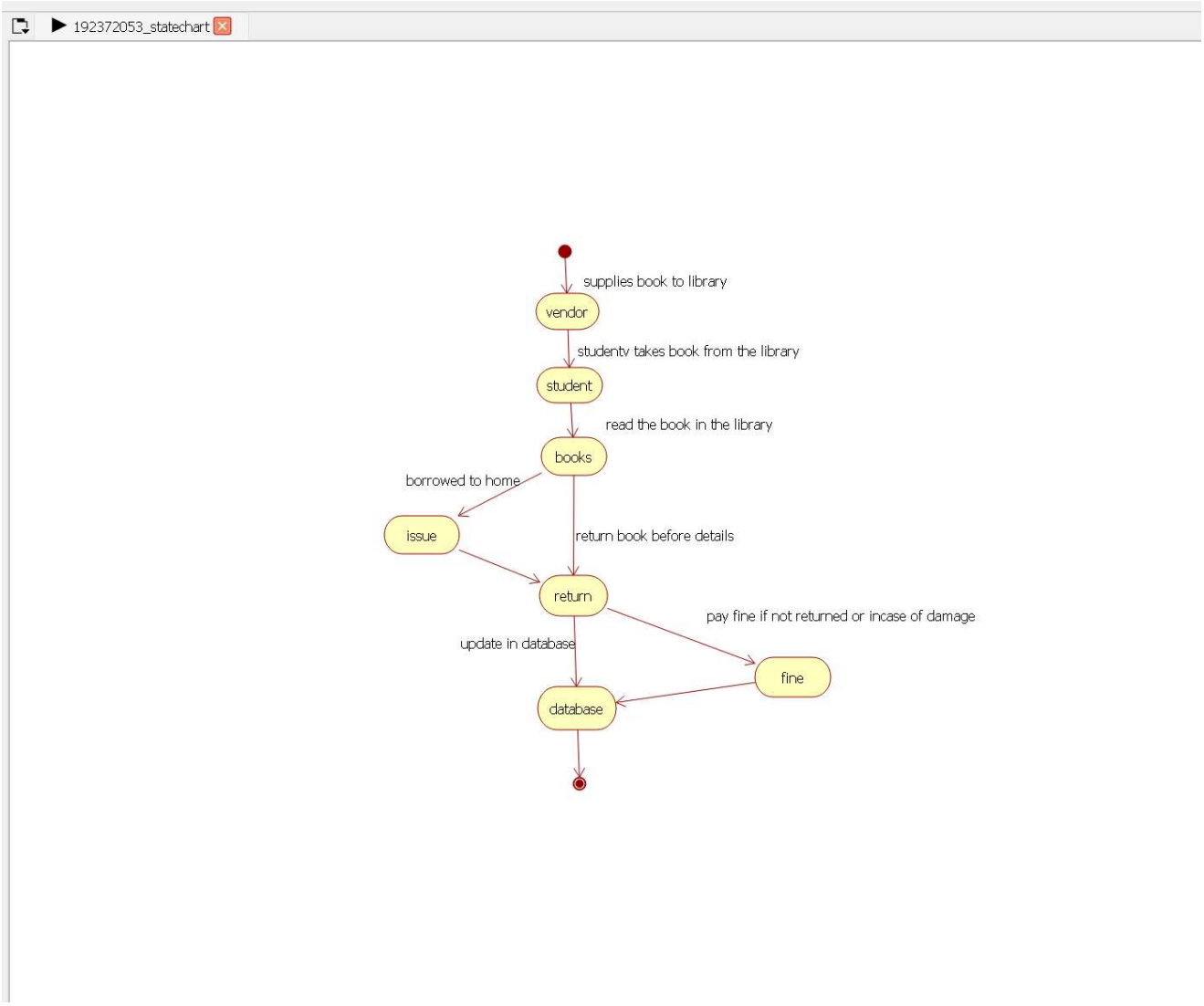
COLLABORATION DIAGRAM:

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



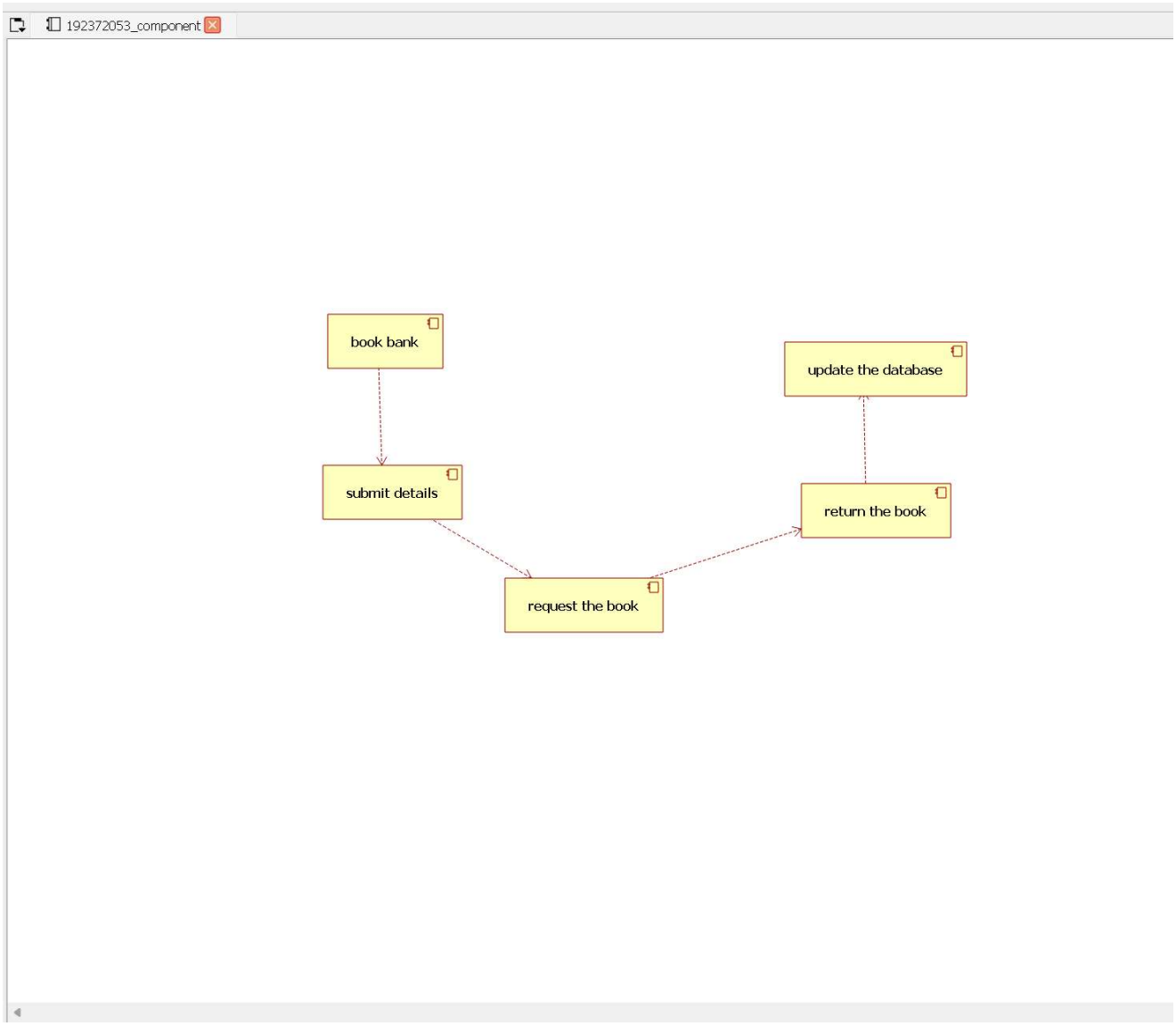
STATE CHART DIAGRAM

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

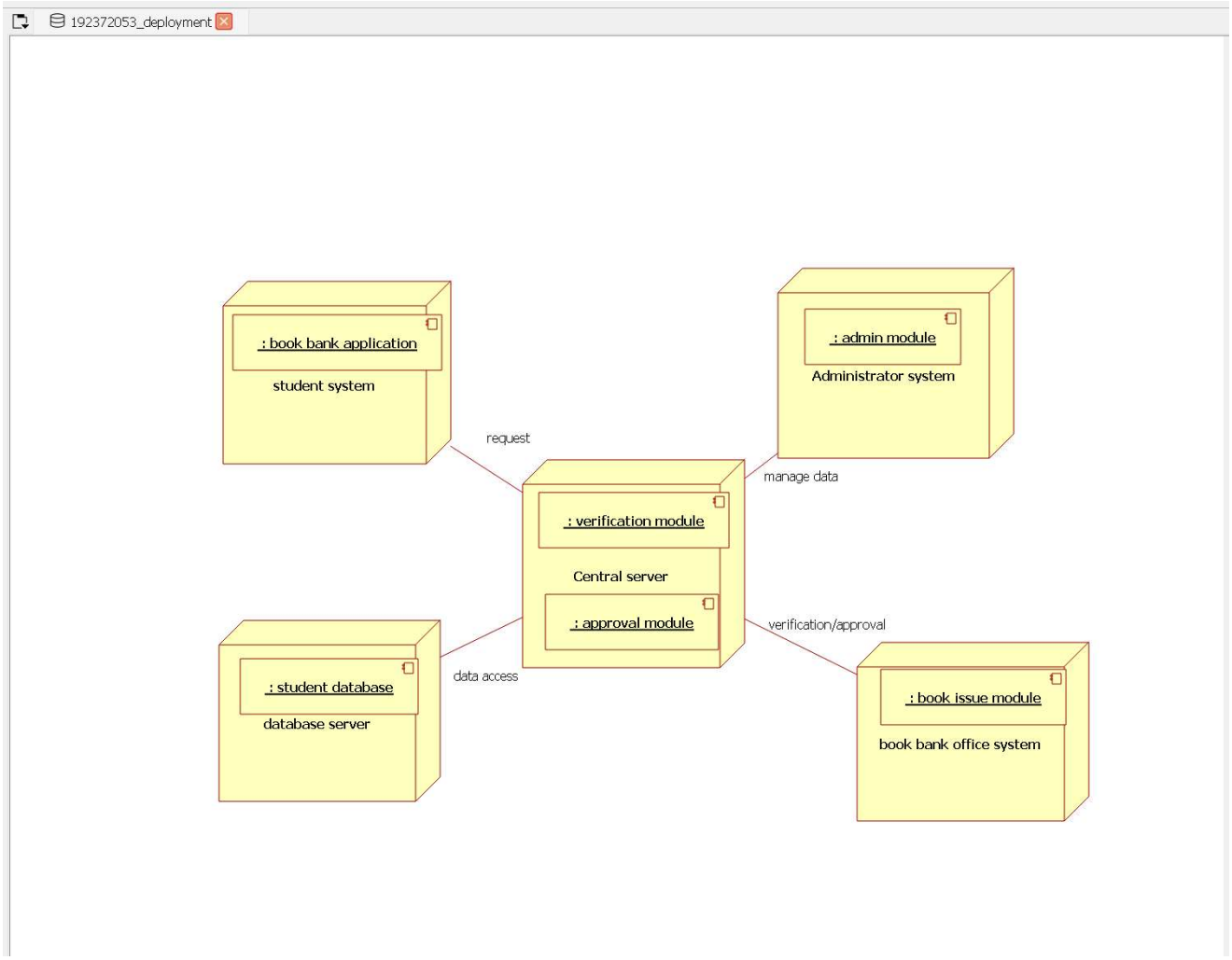
The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by boxed figure. Dependencies are represented by communication association



REGISTER NO:

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

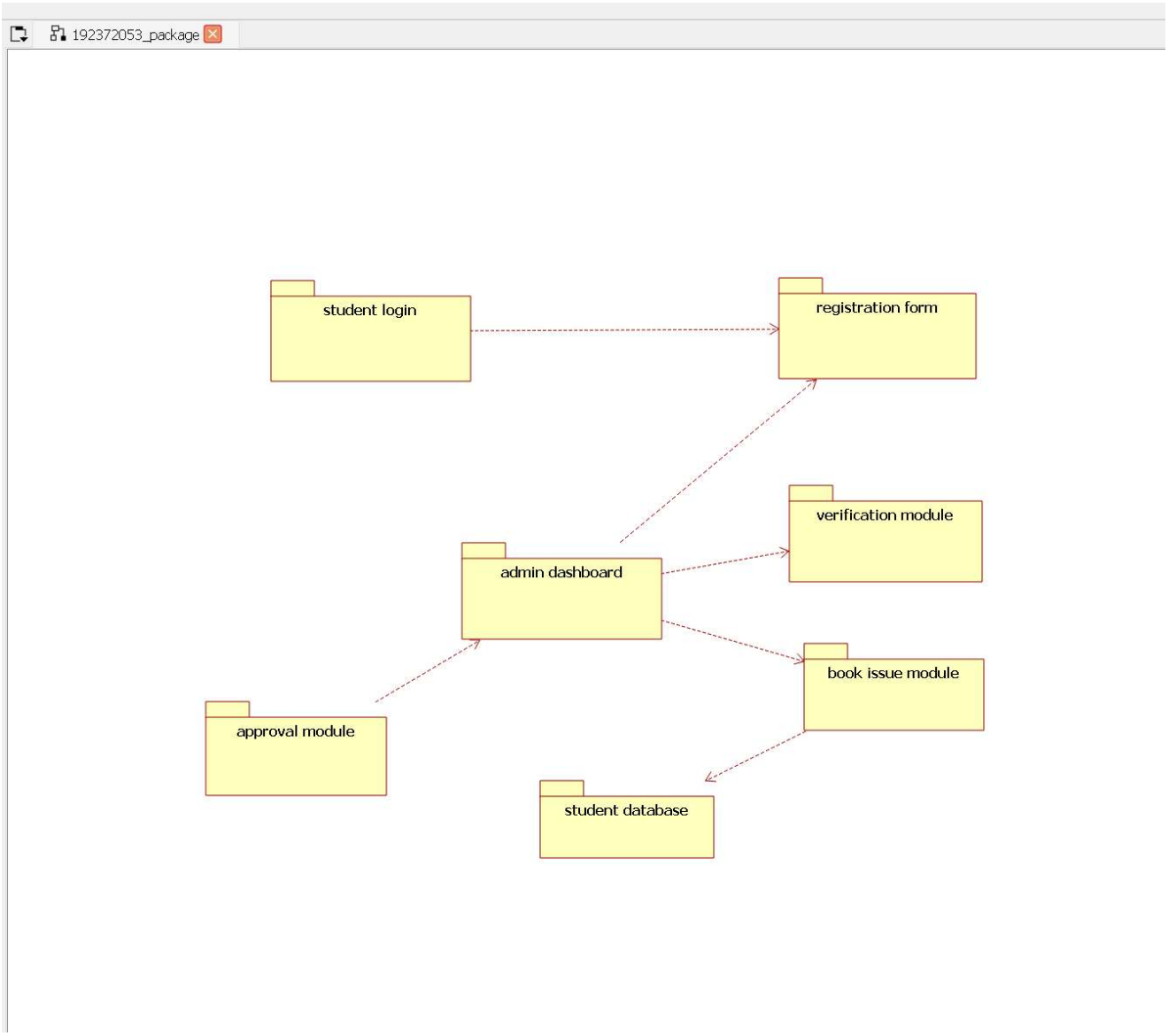


PACKAGE DIAGRAM:

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:

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()

{

}

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()
 {
 }
```



```
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.

EX NO: 3	EXAM REGISTRATION SYSTEM
DATE:	

AIM:

To draw the diagrams [use case, activity, sequence, collaboration, class, statechart, component, deployment, package] for the Exam registration system.

SOFTWARE REQUIREMENTS SPECIFICATION

	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

The Exam Registration is an application in which applicant can register themselves for the exam. The details of the students who have registered for the examination will be stored in a database and will be maintained. The registered details can then be verified for any fraudulent or duplication and can be removed if found so. The database which is verified can be used to issue
hall tickets and other necessary materials to the eligible students.

1.3 PROJECT DESCRIPTION:

This software is designed for the verification of the details of the candidate by the central computer. The details regarding the candidate will be provided to the central computer through the administrator and the computer will verify the details of candidate and provide approval .Then the hall ticket will be issued from the office to the candidate.

1.4 REFERENCES:

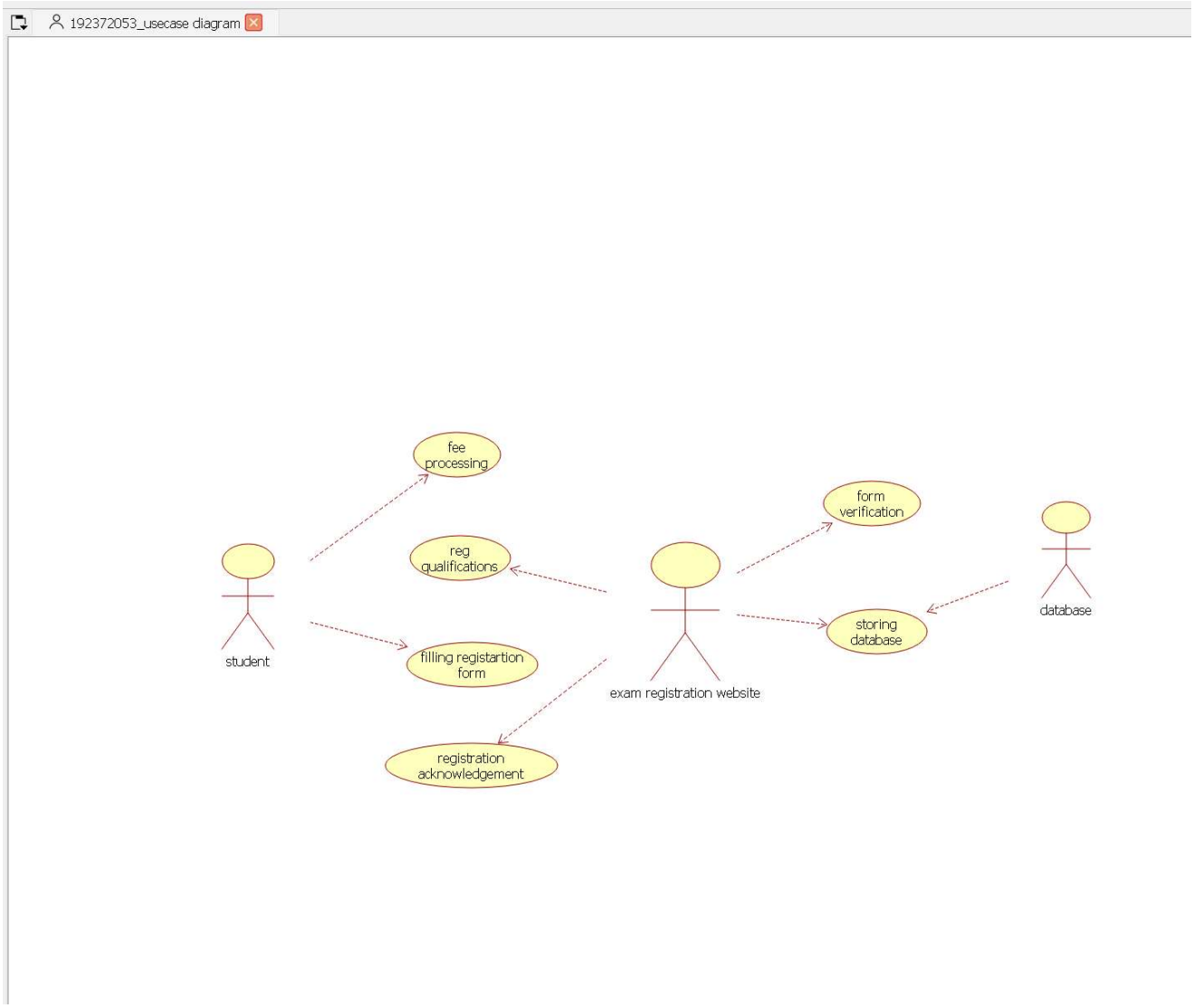
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Student, educational officer..

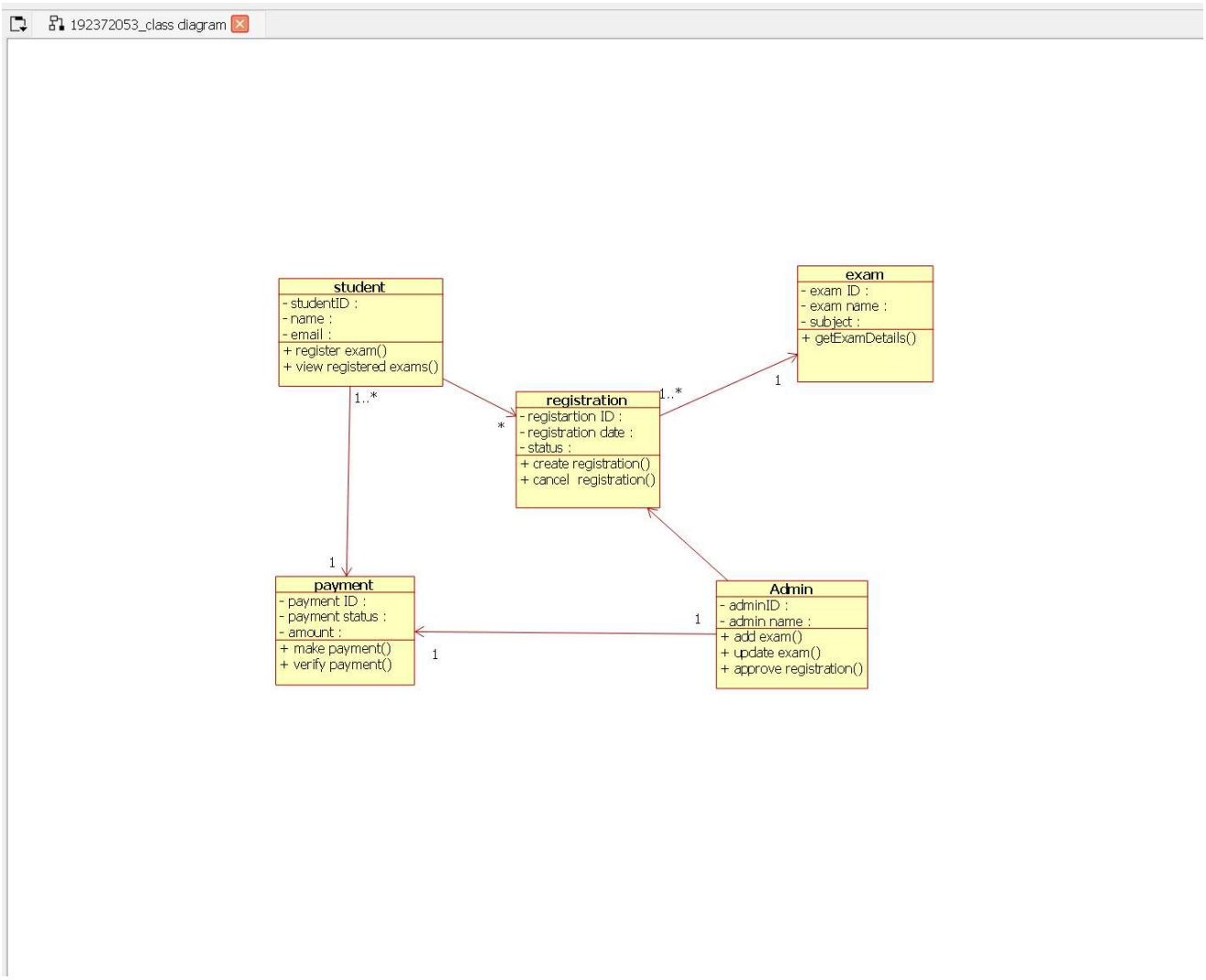
Use case: Student details, student photo, student proof submission of proof ,verification of proof, payment of fees, issue of hall ticket



CLASS DIAGRAM:

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

CLASSES	ATTRIBUTES	OPERATIONS
Central educational system	Student details	Print hall ticket(), Issue hall ticket()
Stud	Submit details, Submit photo	Payment of fees()
Edu officer	Enter details	Issue hall ticket(), Verify proof()

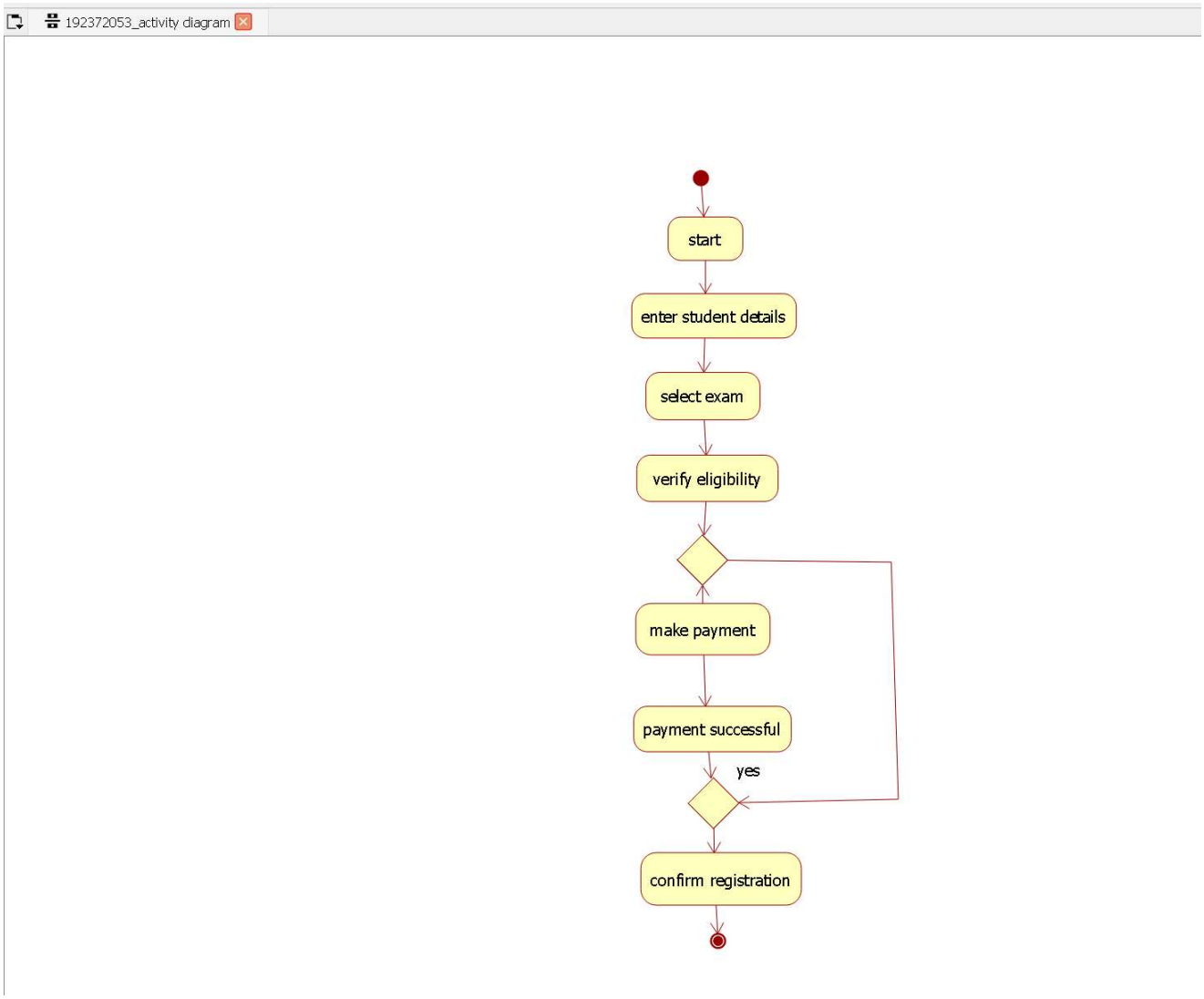


ACTIVITY DIAGRAM:

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

Activities: Enter student details , submit student proof and photo, payment of fees, issue of hall ticket.

Decision box: Verification of proof.

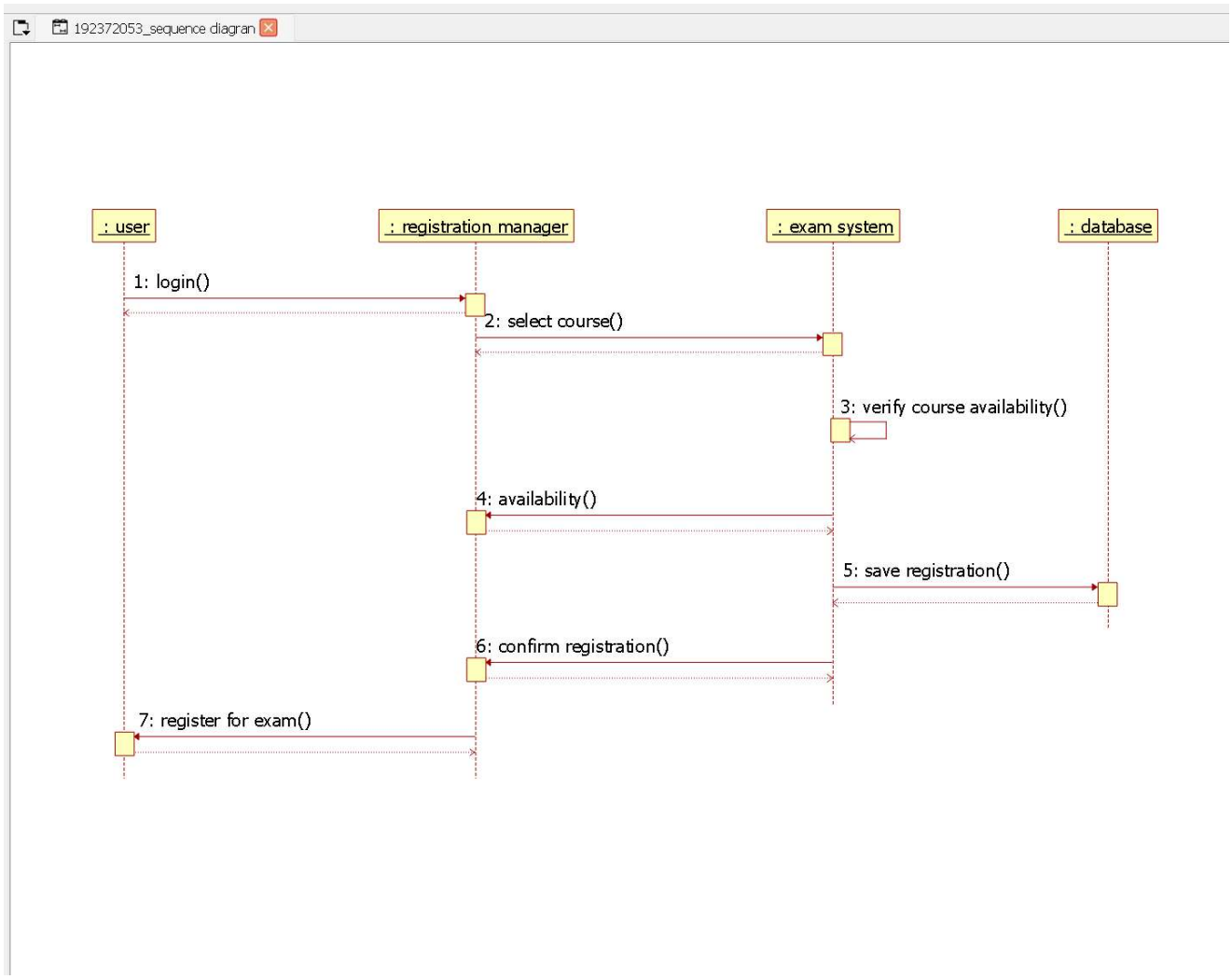


REGISTER NO:

SEQUENCE DIAGRAM:

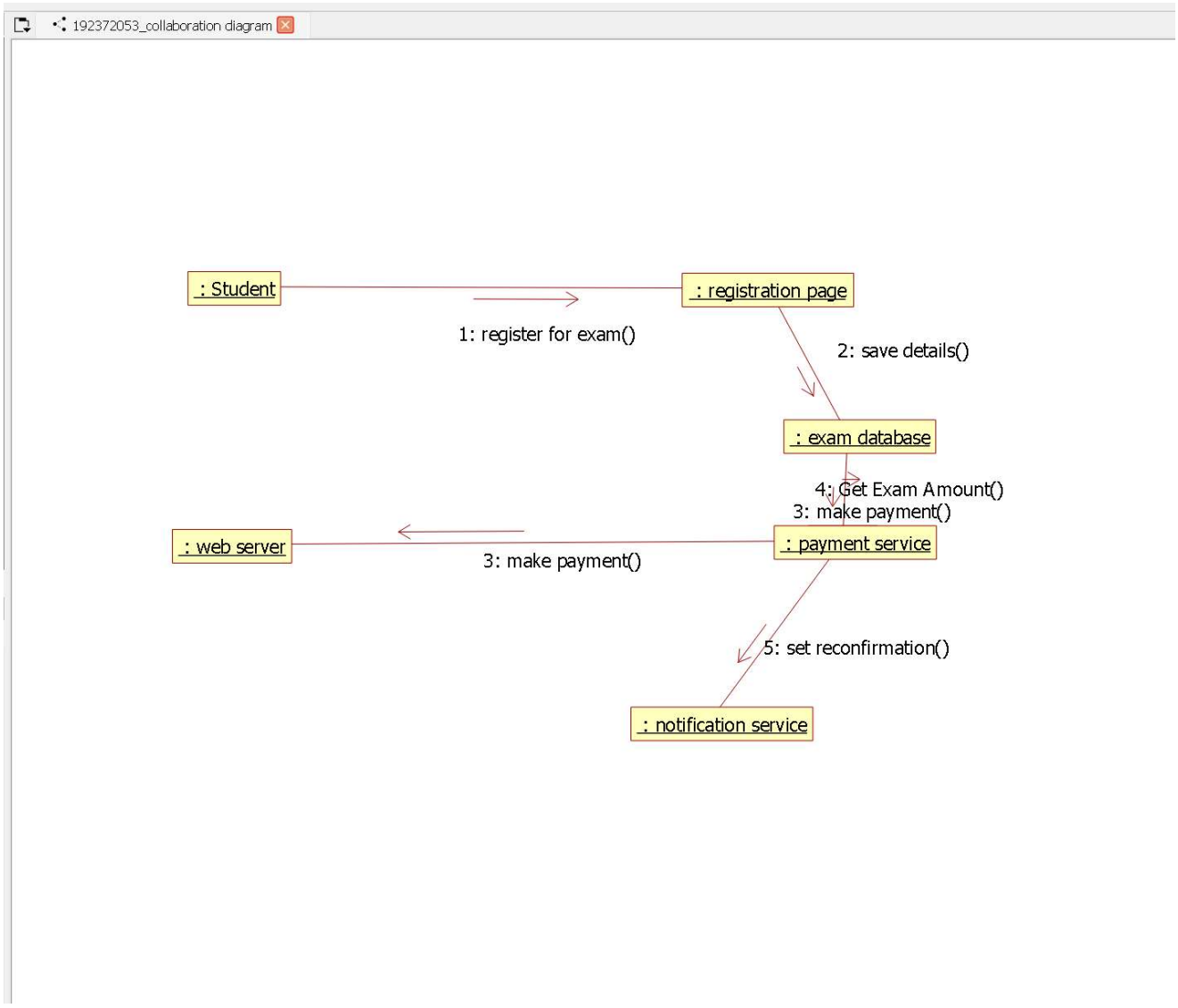
This diagram consists of the objects, messages and return messages.

Object: student, educational officer, central education system.



COLLABORATION DIAGRAM:

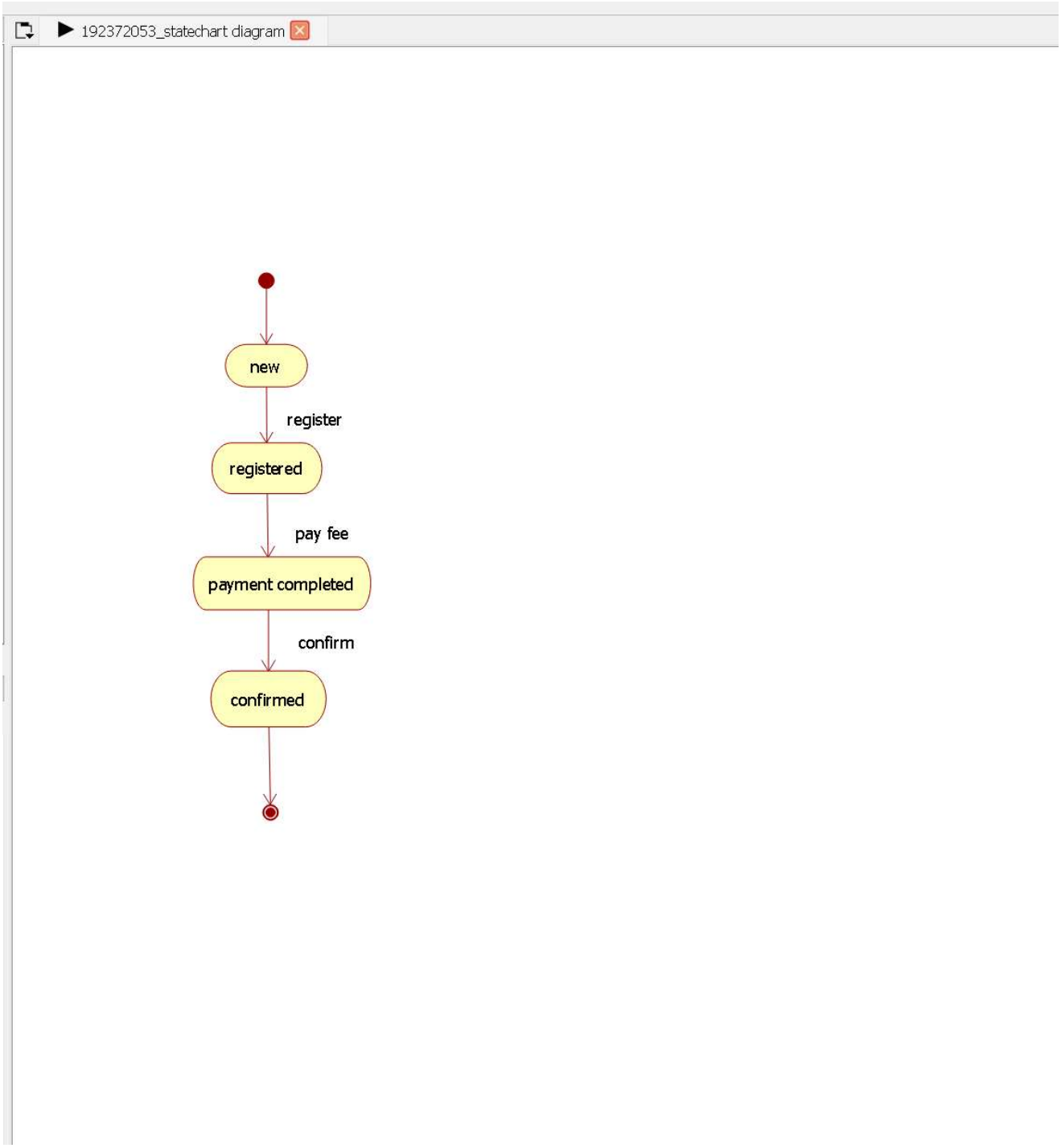
This diagram contains the objects and actors. This will be obtained by the completion of the sequence diagram and pressing the F5 key.



STATE CHART DIAGRAM:

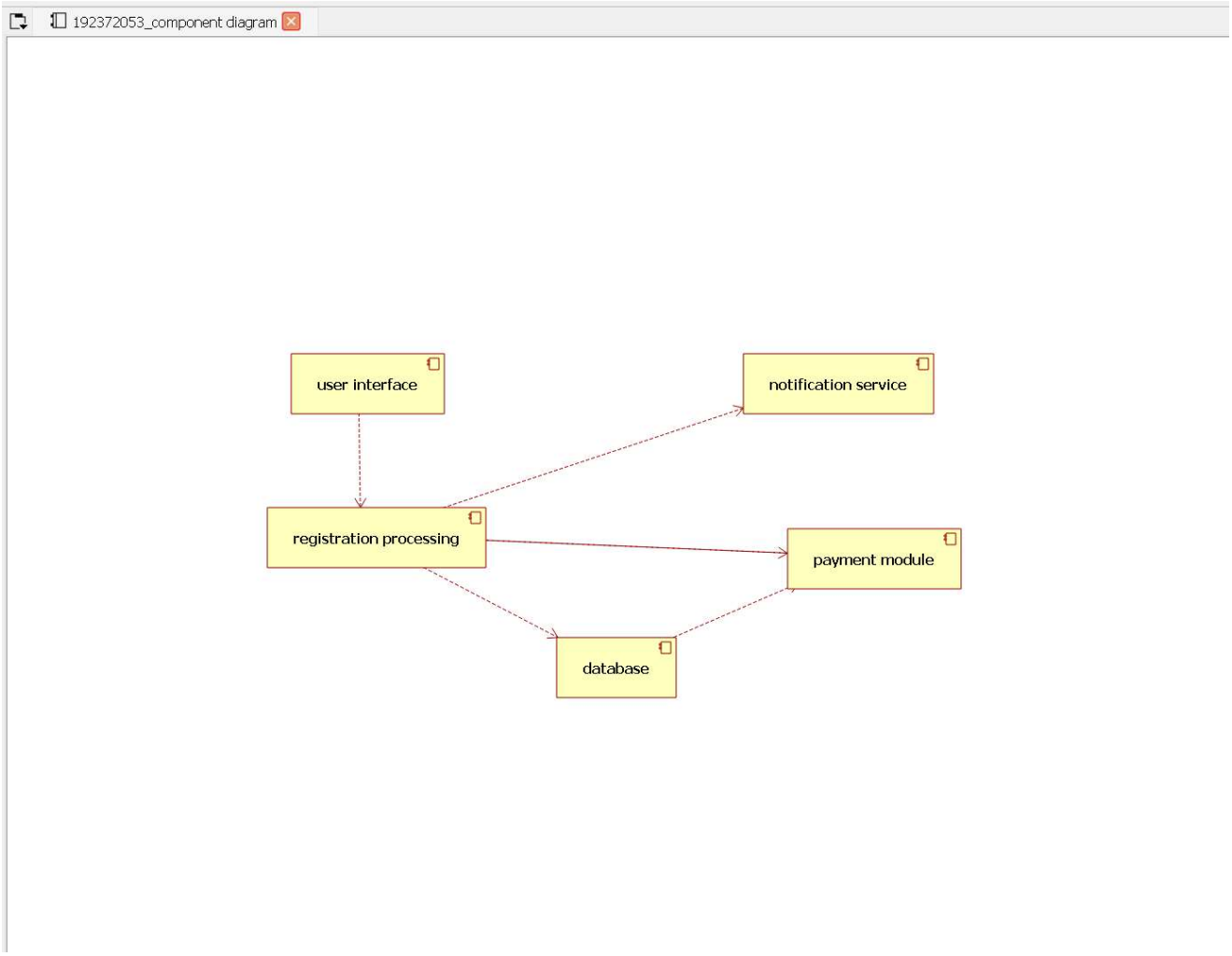
The purpose of state chart diagram is to understand the algorithm involved in performing a method. It is also called as state diagram. A state is represented as a round box, which may contain one or more compartments. An initial state is represented as small dot. A final state is represented as circle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

STATE CHART DIAGRAM:



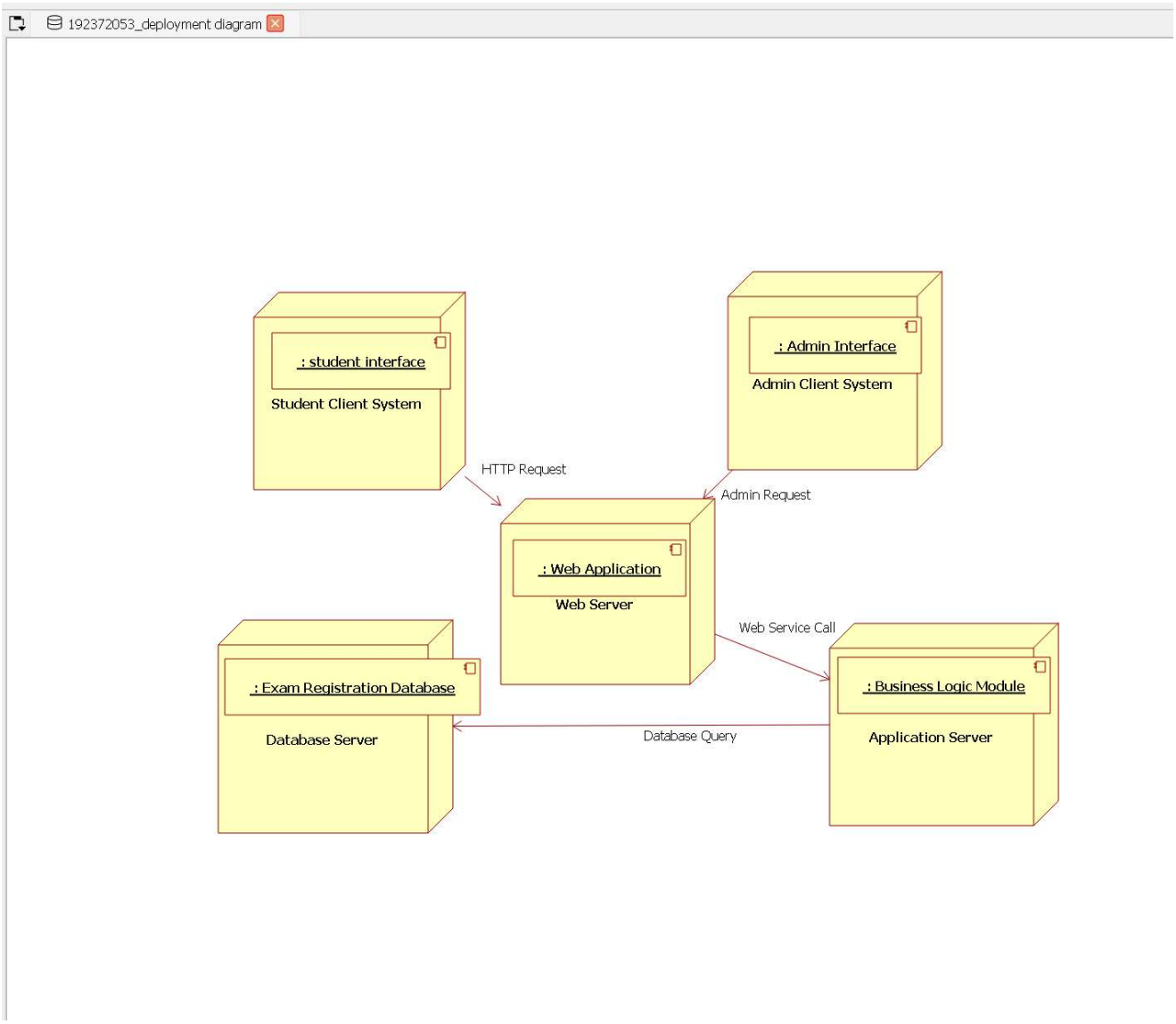
COMPONENT DIAGRAM:

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



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.

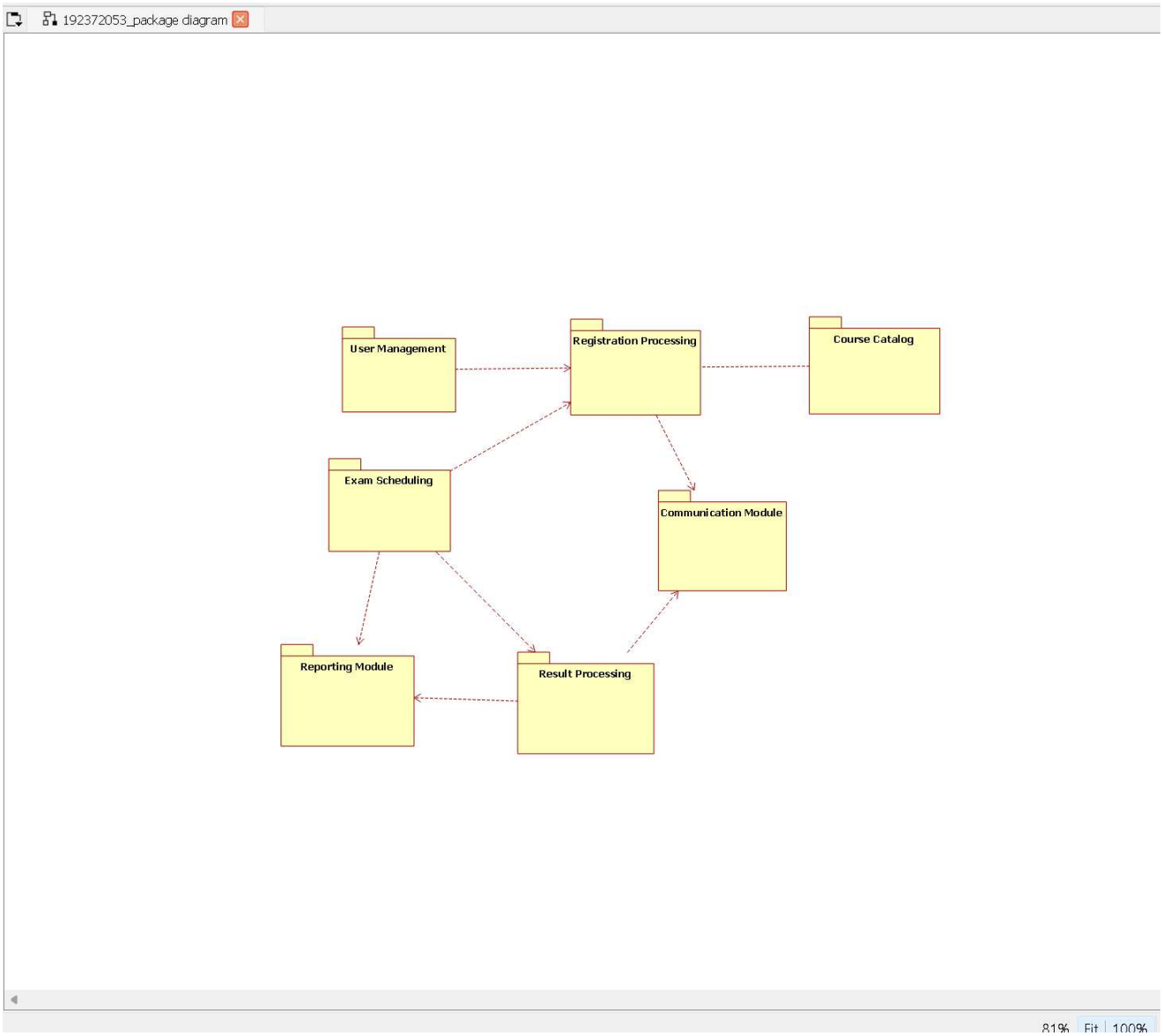


PACKAGE DIAGRAM:

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



REGISTER NO:

PROGRAM CODING:

CENETRAL EDUATIONAL SYSTEM:

```
Public class central educational system {
        Public integer student details;
        Public void valid proof()
         {
         }
}
```

EDUCATIONAL OFFICER:

Public class educational officer

```
{
        Public integer id no;
        Public string name;
        Public void verification of
        proof() {
        }
        Public void issue hall ticket()
        {
        }
}
```

STUDENT:

```
Public class student
{
        Public integer student details;
        Public void payment of fees()
        {
```



```
        }
        Public void receive hall ticket()
        {
        }
}
```

RESULT:

Thus the diagrams [use case, activity, sequence, collaboration, class, statechart, component, deployment, package] for the Exam registration system.

EX:NO:4	STOCK MAINTAINANCE SYSTEM
DATE:	

AIM:

To draw the diagrams [usecase, activity, sequence, collaboration, class, collaboration, deployment, state chart , package] for the Stock maintainence system.

SOFTWARE REQUIREMENTS SPECIFICATION:

	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

1.0HARDWARE REQUIREMENTS:

Intel Pentium Processor I3/I5

1.1SOFTWARE REQUIREMENTS:

Rational rose / Argo UML

1.2 PROBLEM ANALYSIS AND PROJECT PLANNING :

The Stock Maintenance System, initial requirement to develop the project about the mechanism of the Stock Maintenance System is caught from the customer. The requirement are

analyzed and refined which enables the end users to efficiently use Stock Maintenance System. The complete project is developed after the whole project analysis explaining about the scope and the project statement is prepared.

1.3PROJECT DESCRIPTION:

This software is designed for supporting the computerized stock maintainence System .In this system, the customer can place order and purchase items with the aid of the stock dealer and central stock system. This orders are verified and the items are delivered to the customer. **1.4 REFERENCES:**

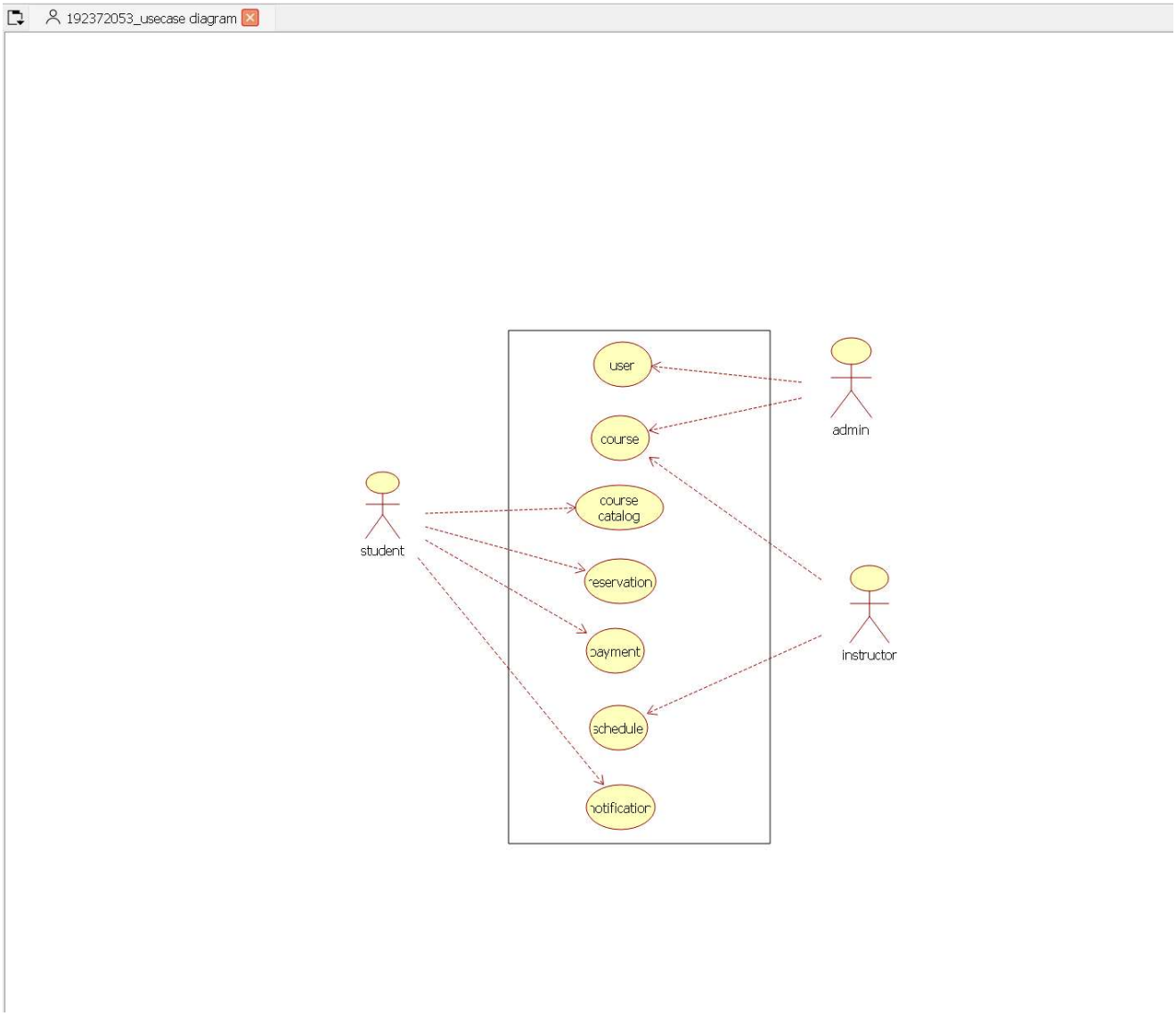
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Customer, Stock dealer, central stock system.

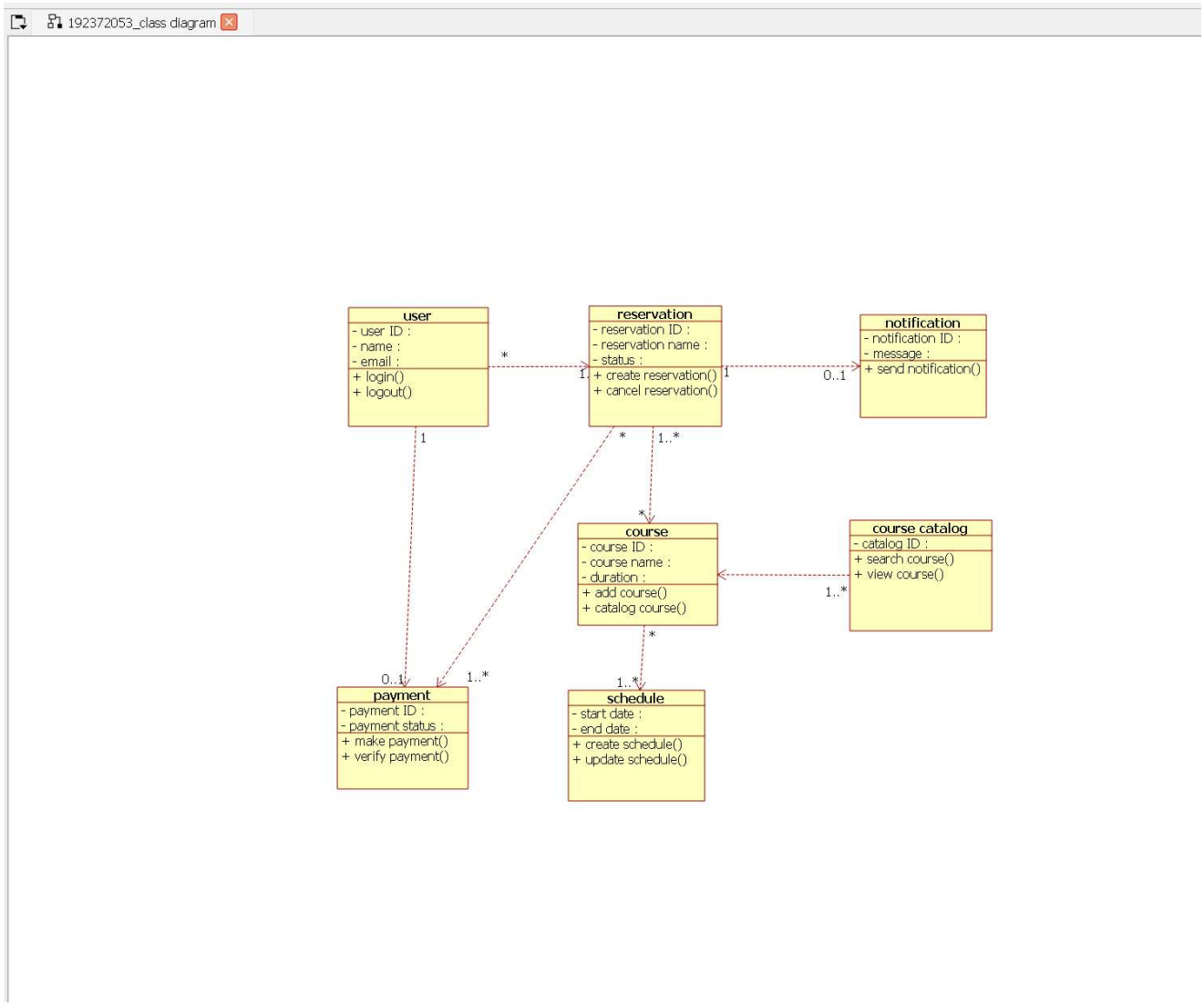
Use case: purchase order, verification of order, payment ,delivery of items.



CLASS DIAGRAM:

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

CLASSES	ATTRIBUTES	OPERATIONS
Central stock system	Store stock details	Print bill()
Stock dealer	Take order	Deliver item()
Customer	Place order	Payment()

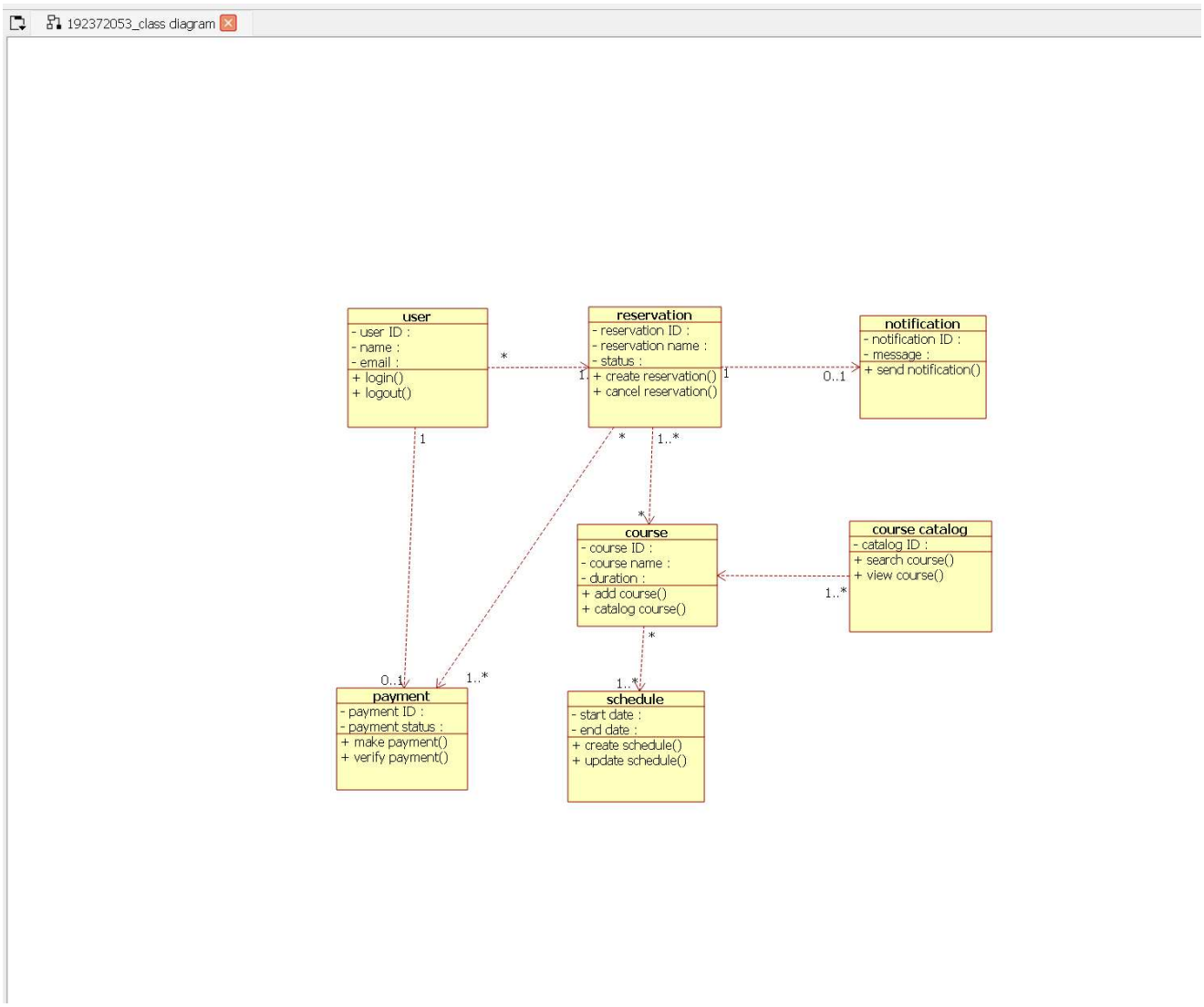


ACTIVITY DIAGRAM:

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

Activities: Purchase order, payment , delivery of items.

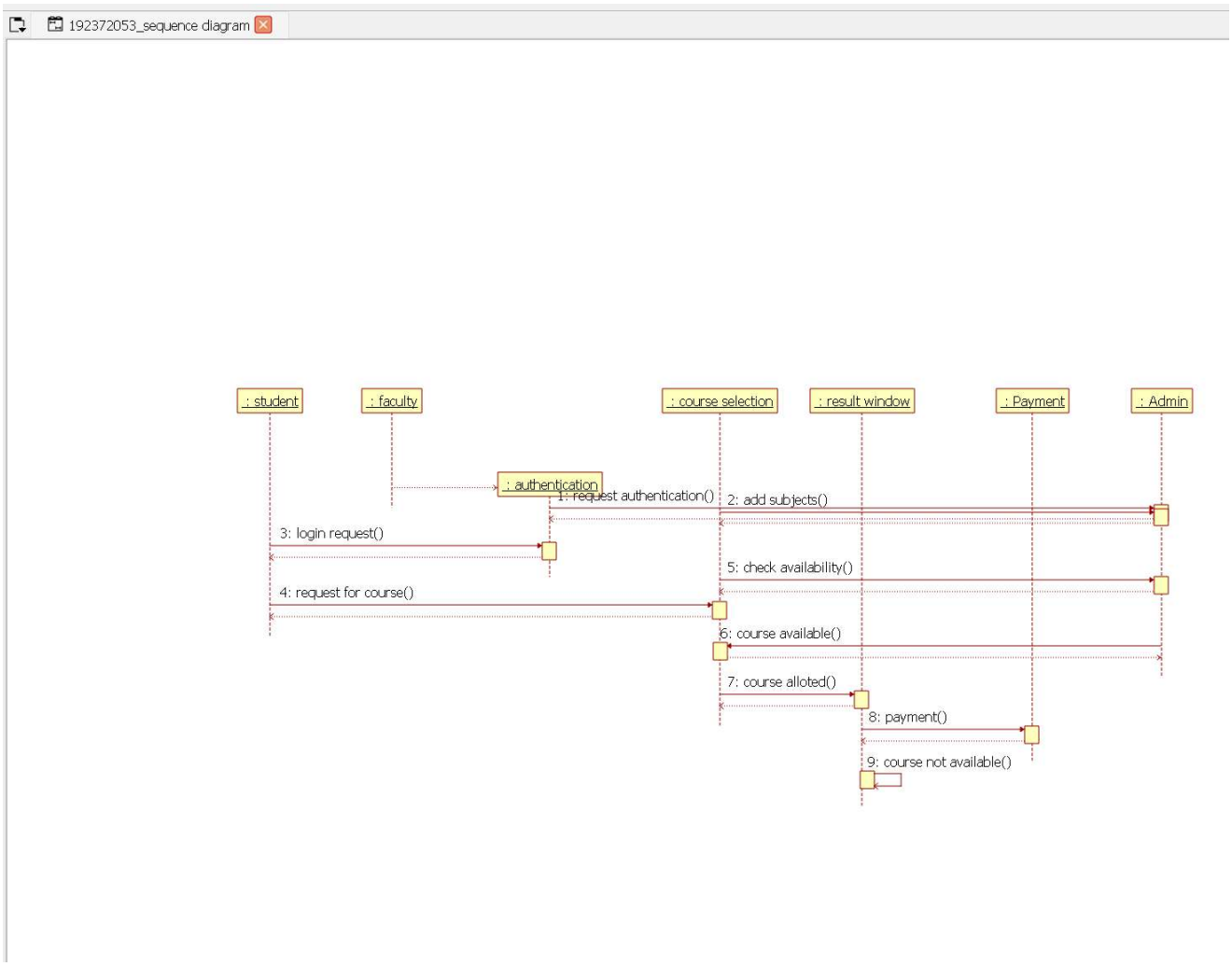
Decision box: Valid or not



SEQUENCE DIAGRAM:

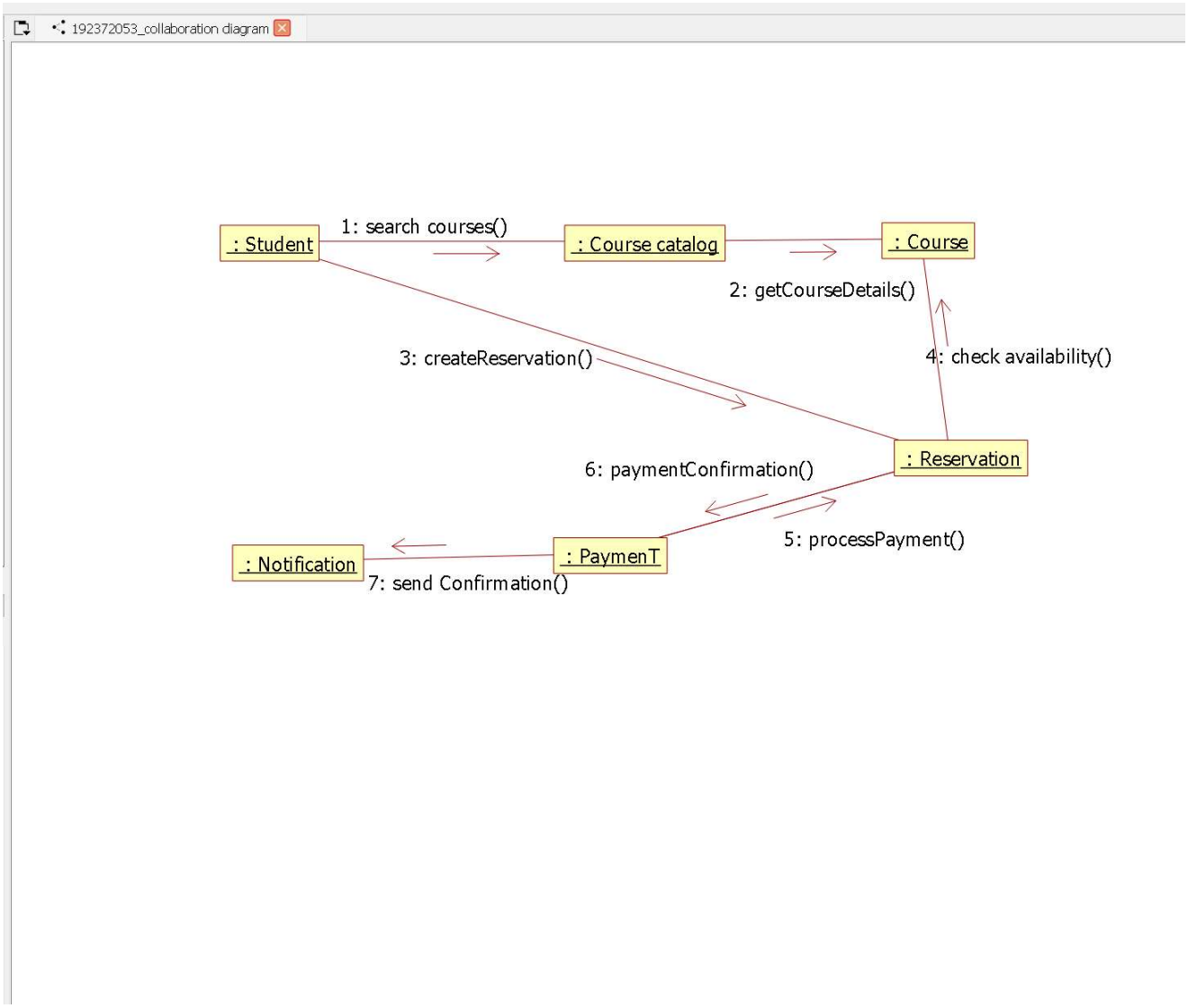
This diagram consists of the objects, messages and return messages.

Object: Customer, Stock dealer, Central stock system



COLLABORATION DIAGRAM:

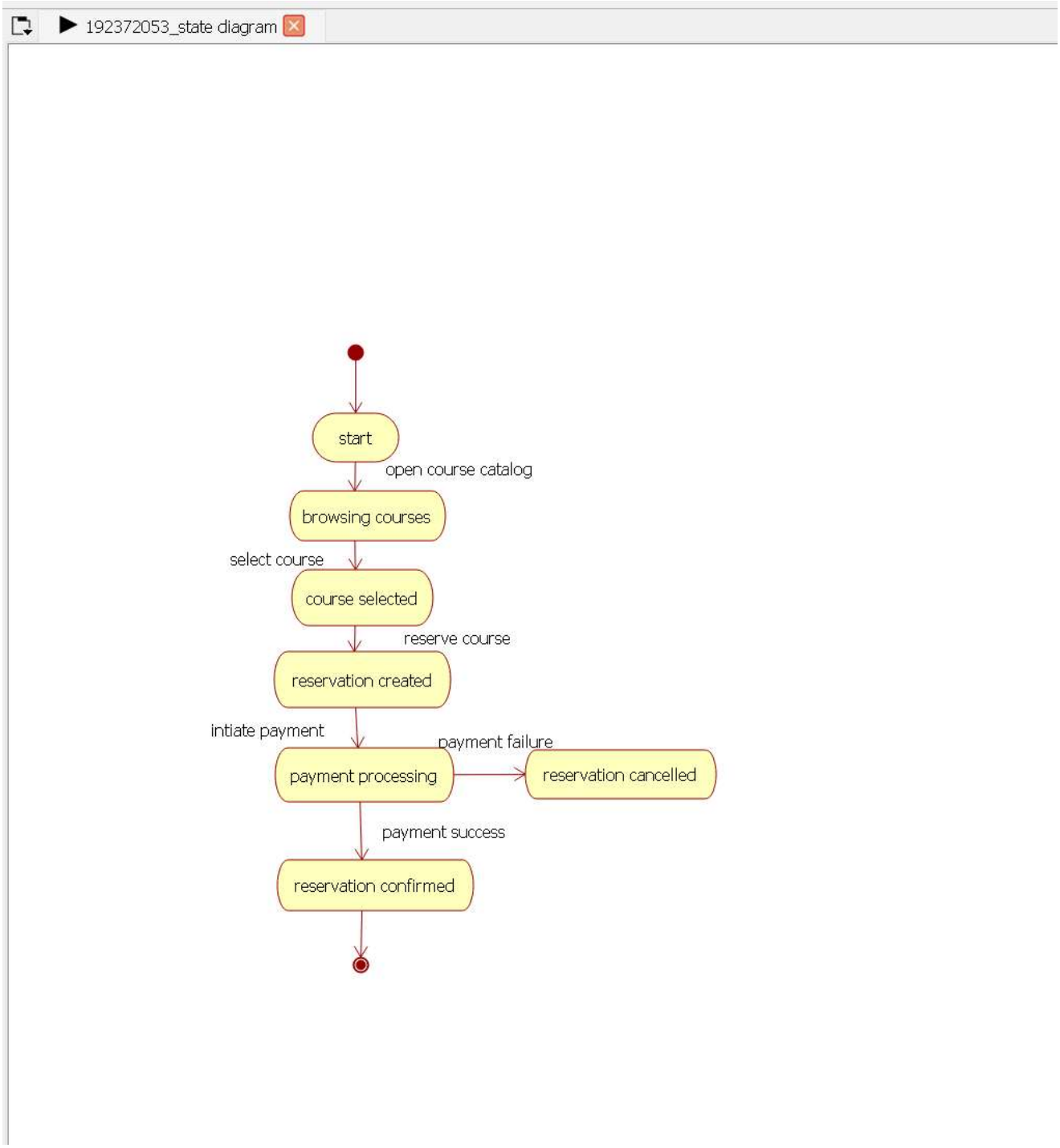
This diagram contains the objects and actors. This will be obtained by the completion of the sequence diagram and pressing the F5 key.



REGISTER NO:

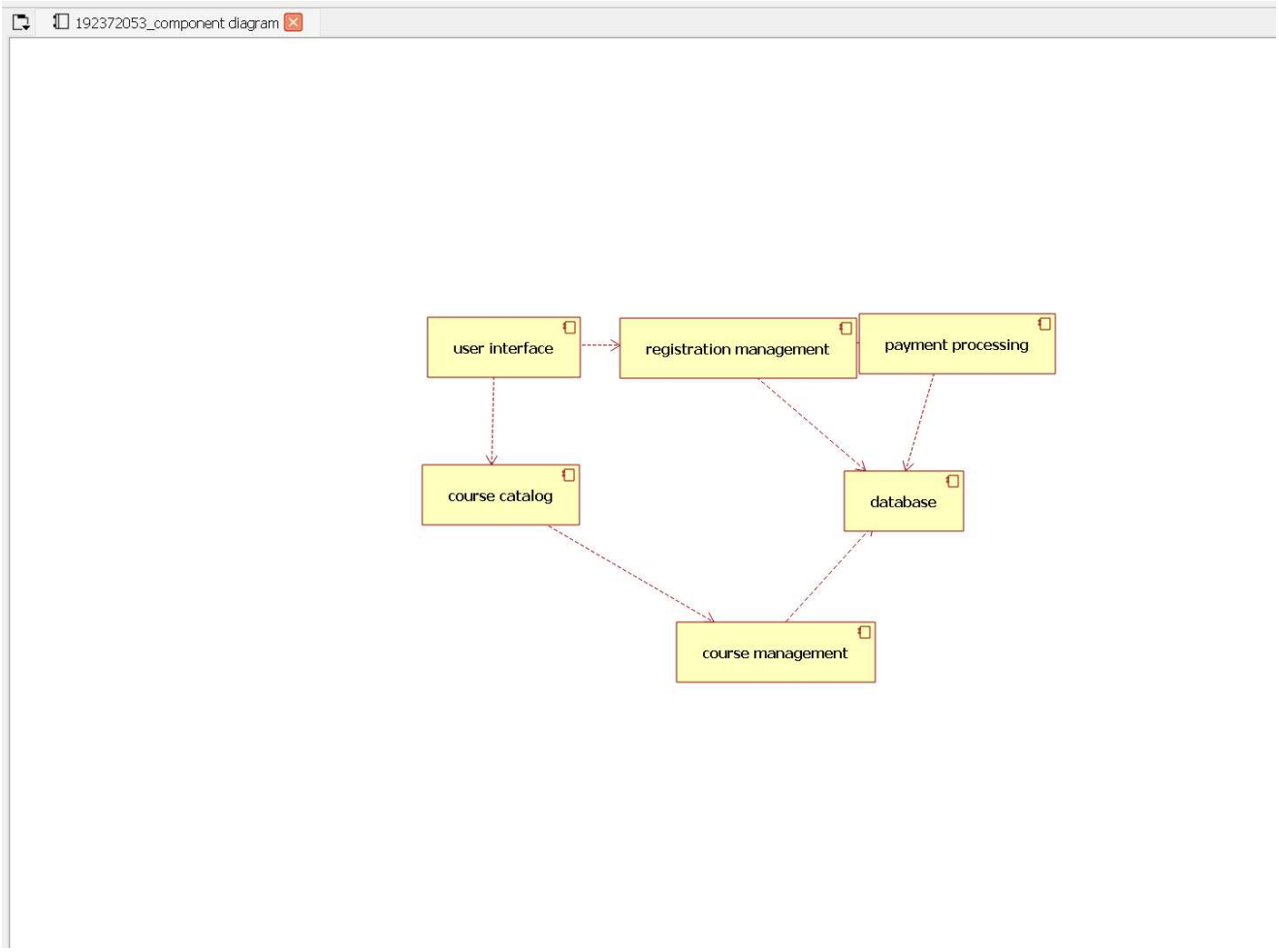
STATE CHART DIAGRAM:

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. A final state is represented as circle surrounding a small dot.



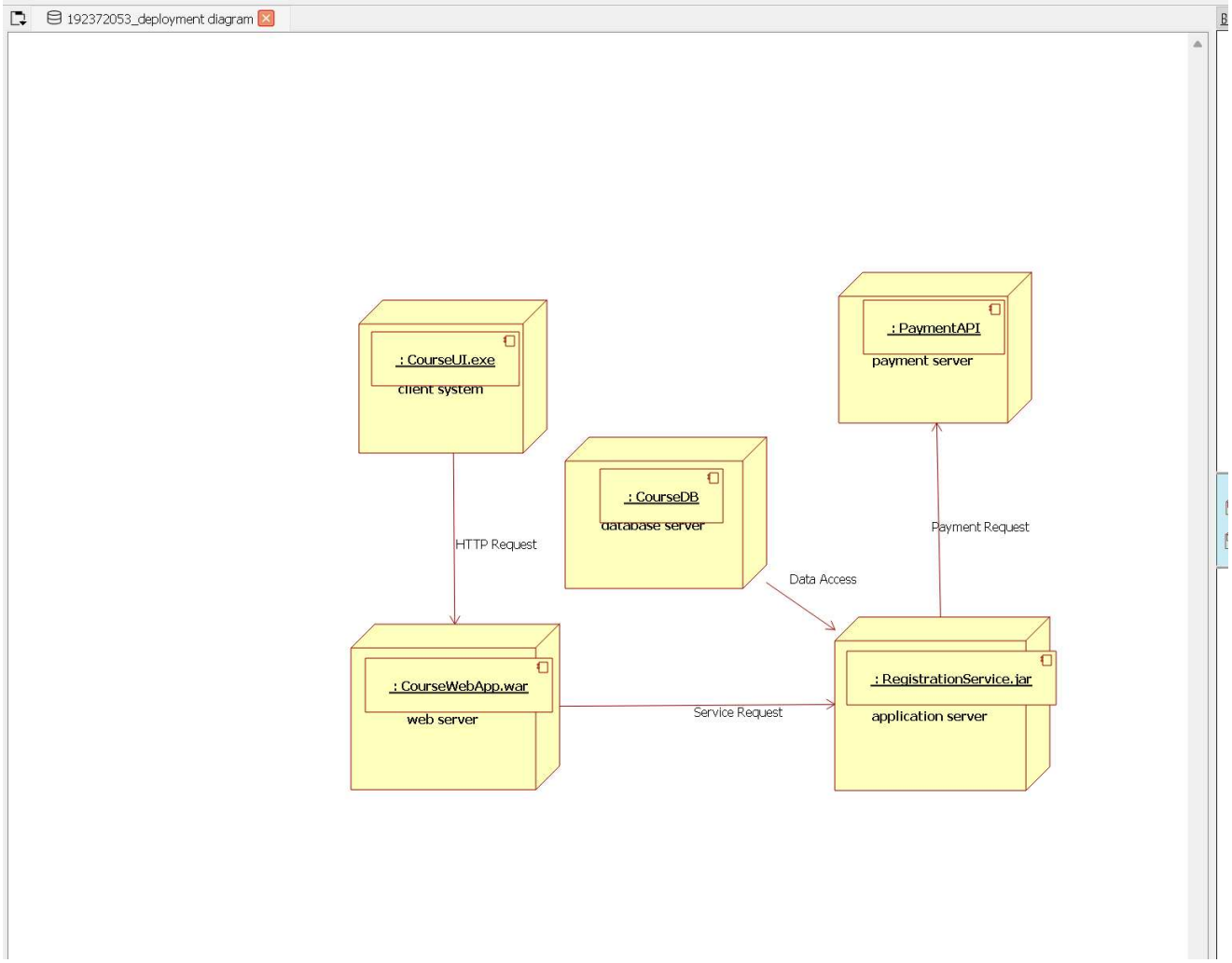
COMPONENT DIAGRAM:

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



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.

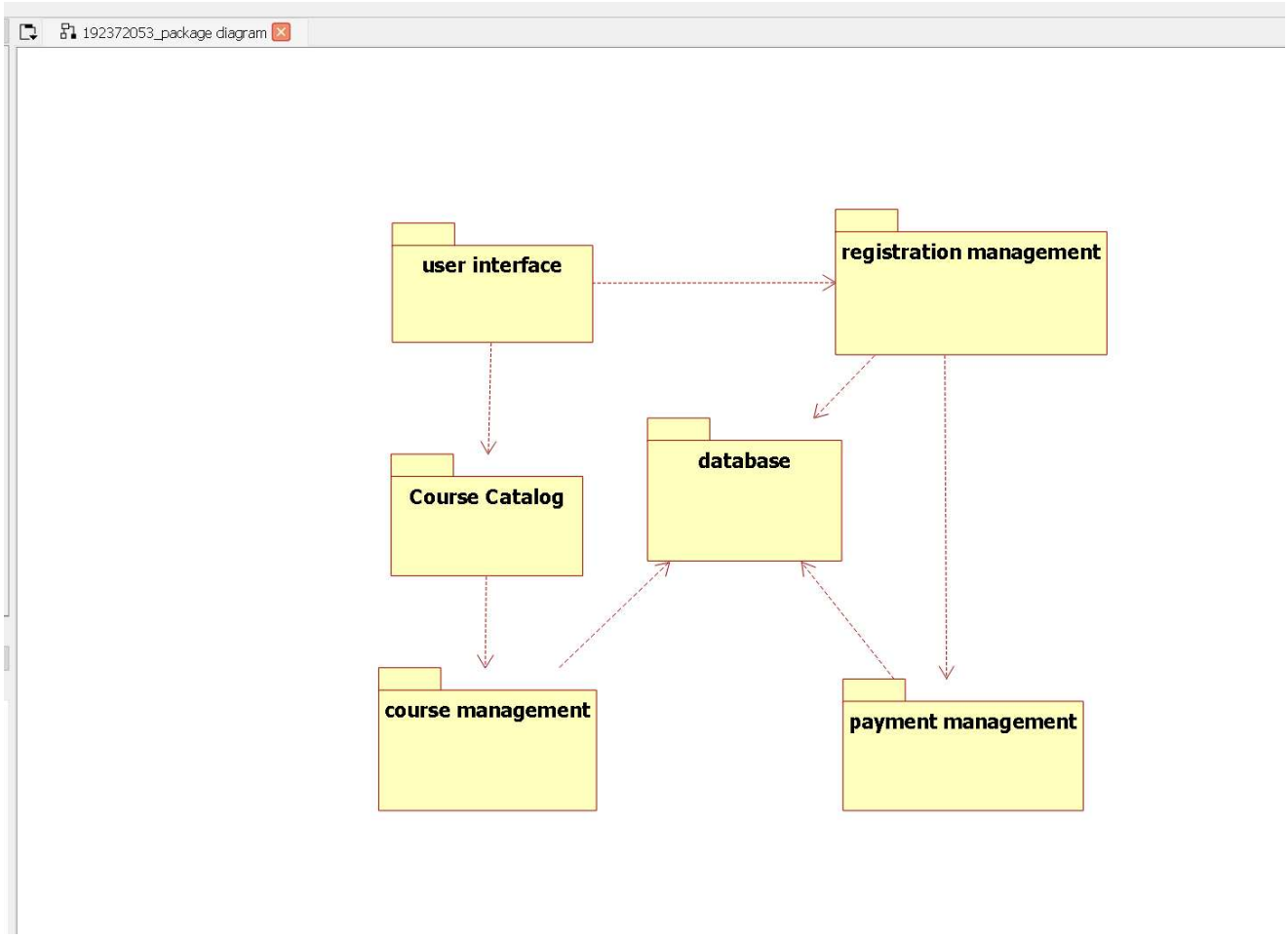


PACKAGE DIAGRAM:

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:

CENTRAL STOCK SYSTEM:

Public central stock system

{

Public integer store stock details;

Public void print bill()

{

}

Public void deliver product()

{

}

}

CUSTOMER:

```
Public class customer
{
        Public integer place order;
        Public void payment()
         {
         }
}
```

STOCK DEALER:

```
Public class stock dealer
{
        Public integer take order;
        Pubic integer enter details;
        Public integer verify details;
        Public void deliver item()
         {
         }
}
```

RESULT:

Thus the diagrams [usecase, activity, sequence, collaboration, class, collaboration, deployment, compoenent, statechart, package] for the Stock maintainence system.

REGISTER NO:

EX NO:5	ONLINE COURSE RESERVATION SYSTEM

DATE:	

AIM:

To draw the diagrams [usecase, activity, sequence, collaboration, class, statechart, component, deployment, package] for the Online course reservation system.

SOFTWARE REQUIREMENTS SPECIFICATION:

	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:

The requirement form the customer is got and the requirements about the course registration are defined. The requirements are analyzed and defined so that is enables the student to efficiency select a course through registration system. The project scope is identified and the problem statement is prepared.

1.3PROJECT DESCRIPTION:

This software is designed for supporting online course reservation system. This system is organized by the central management system . The student first browses and select the desired course of their choice. The university then checks the availability of the seat if it is available the student is enrolled for the course.

1.4 REFERENCES:

IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

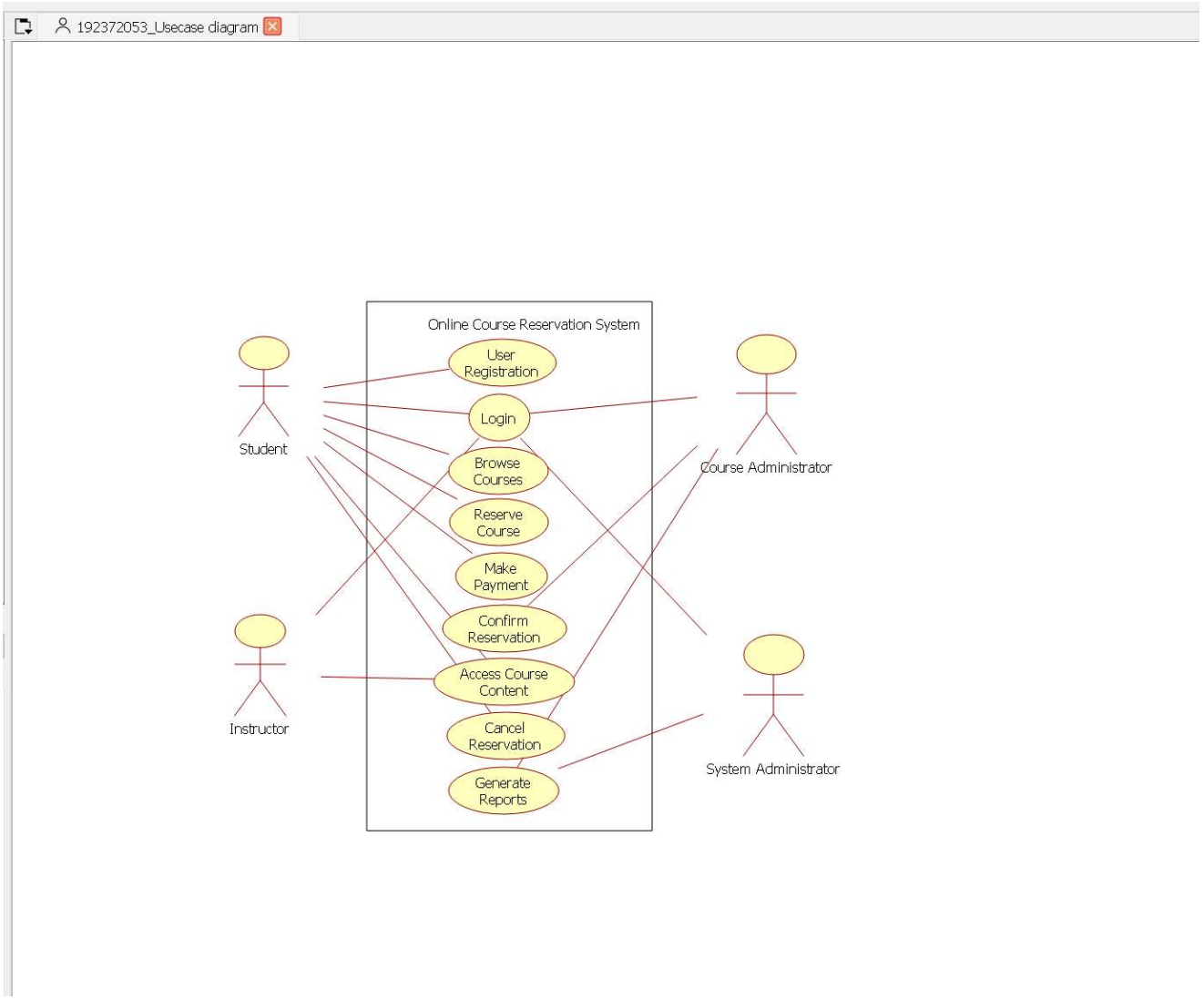
OOAD LAB

REGISTER NO:

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

Actors: Student, University.

Use case: Browse course, select course, register, submit details, verify details, pay fees, enroll student.

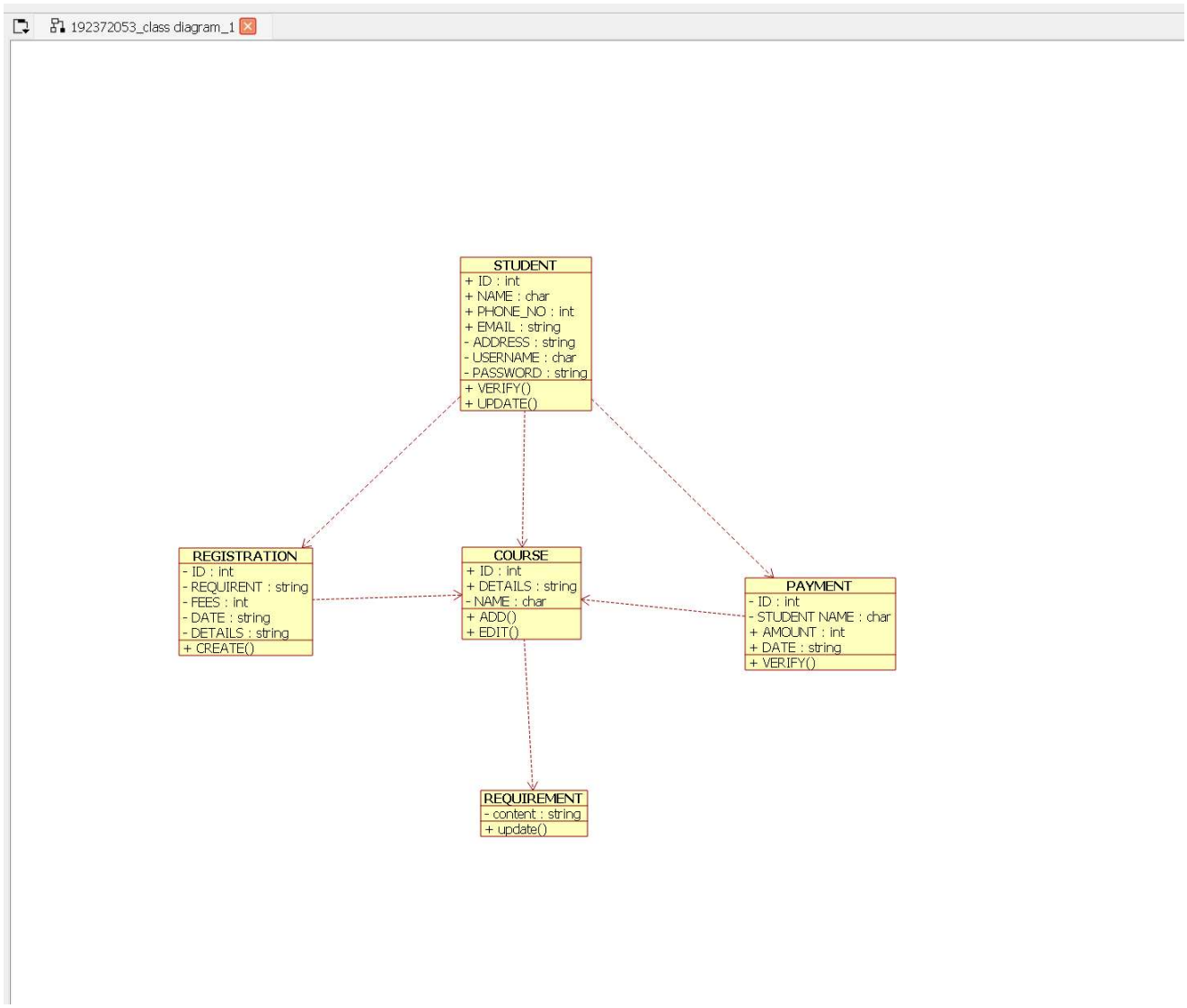


CLASS DIAGRAM:

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

CLASSES	ATTRIBUTES	OPERATIONS
Central management system	Store details	Verify()
Student	Name and address	Browse()

University	Store details	Verify()

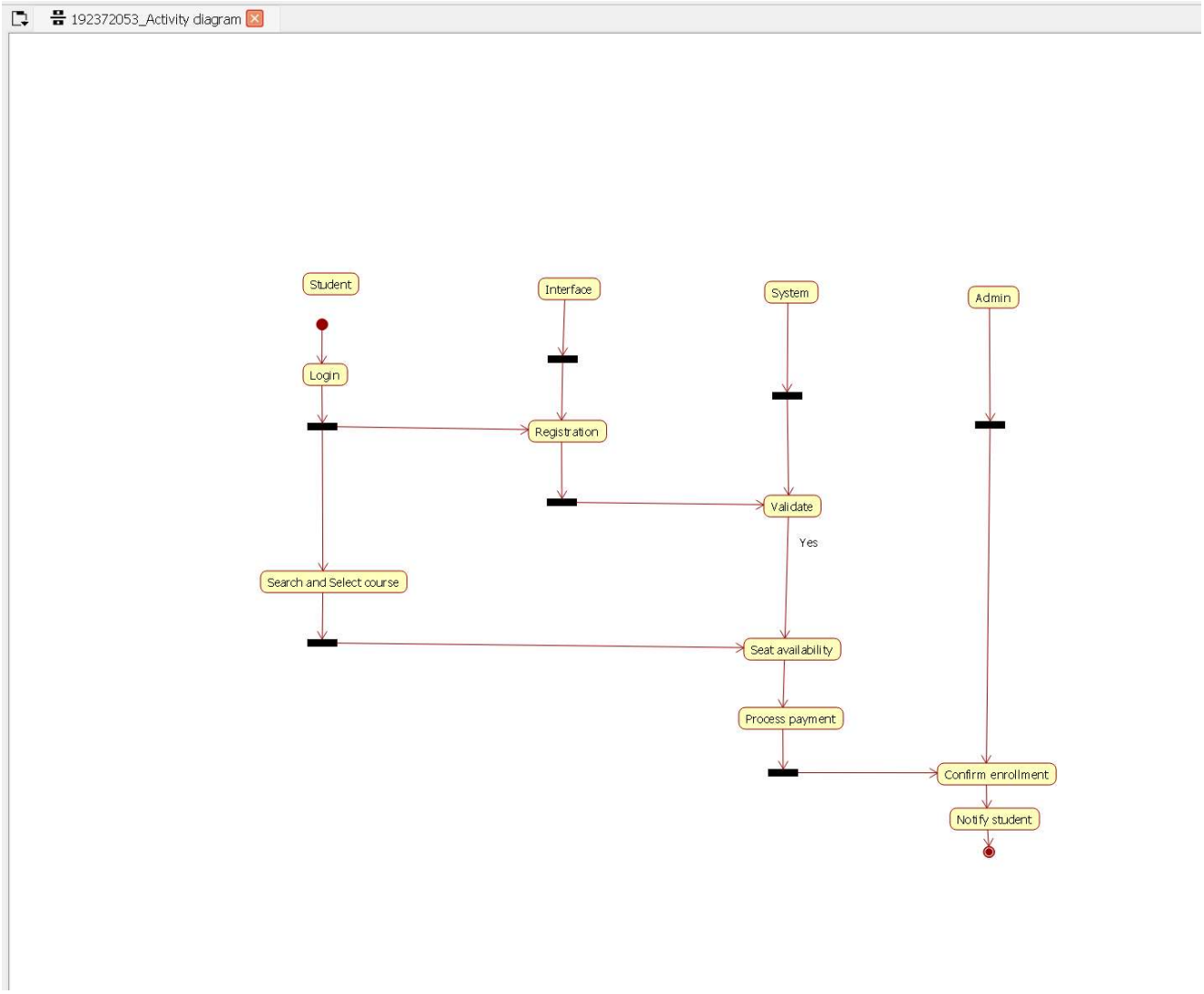


ACTIVITY DIAGRAM:

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

Activities: Browse course, select course, register course, submit details

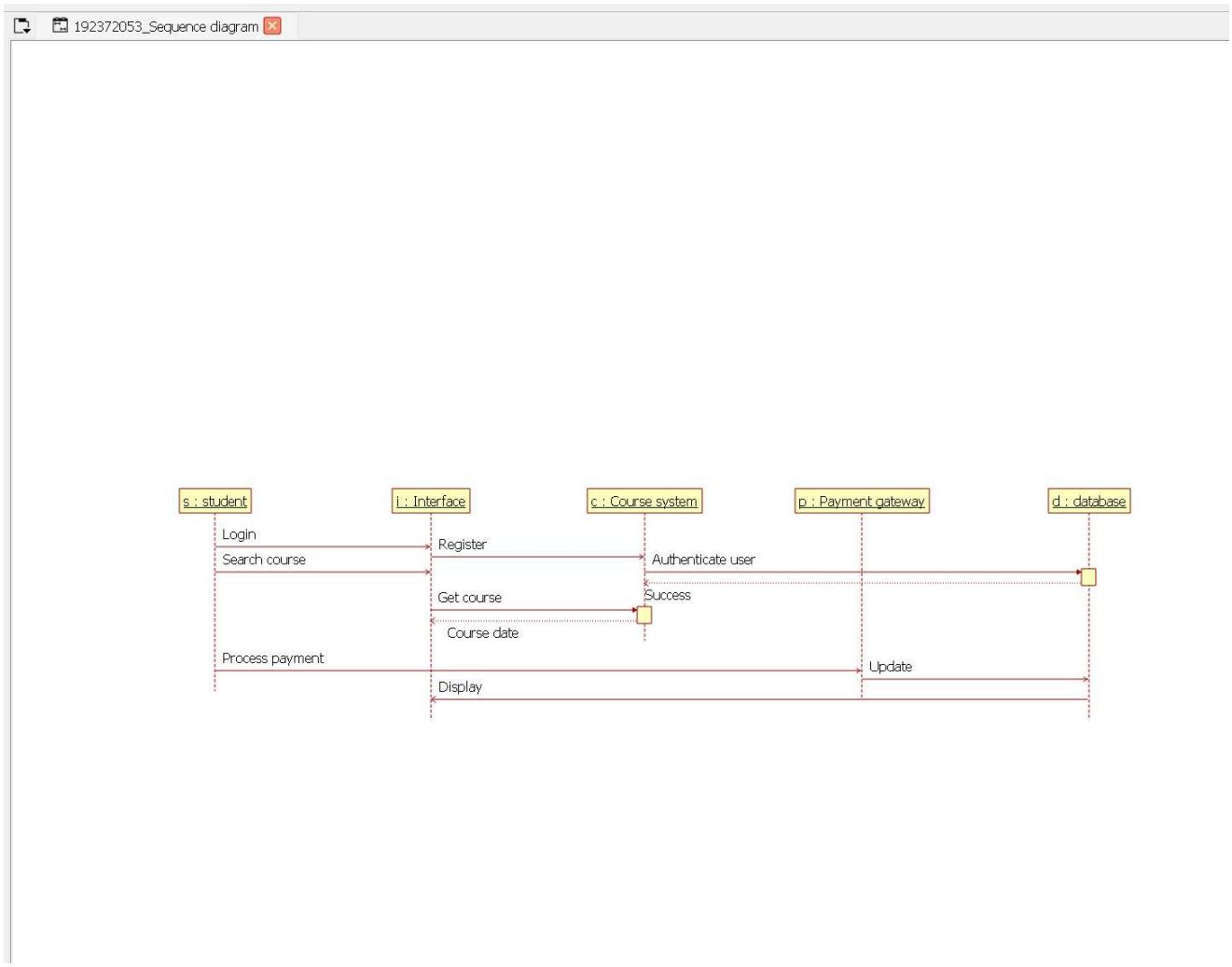
Decision box: check availability or not.



SEQUENCE DIAGRAM:

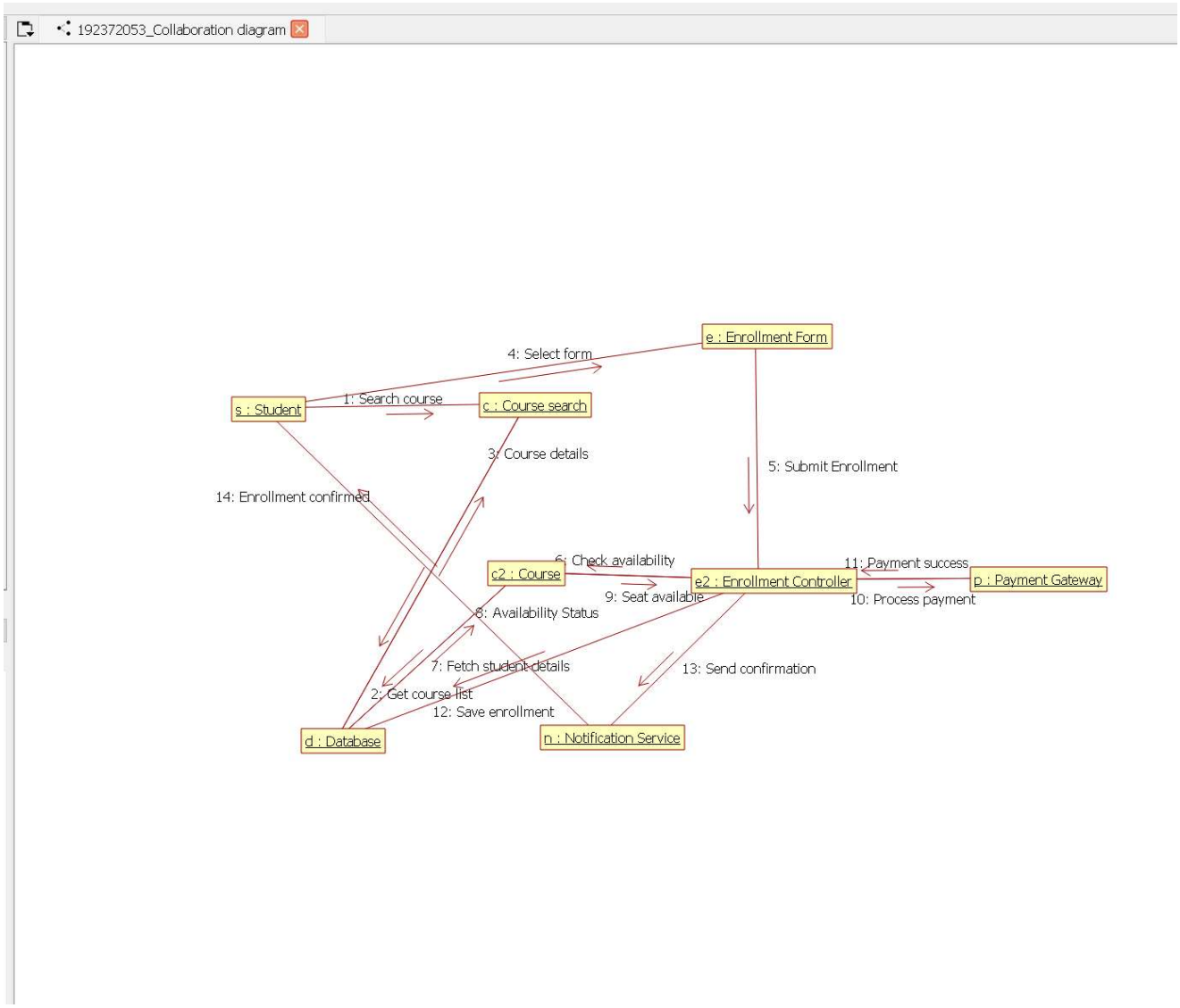
This diagram consists of the objects, messages and return messages.

Object: Student, University, Central management system



COLLABORATION DIAGRAM:

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

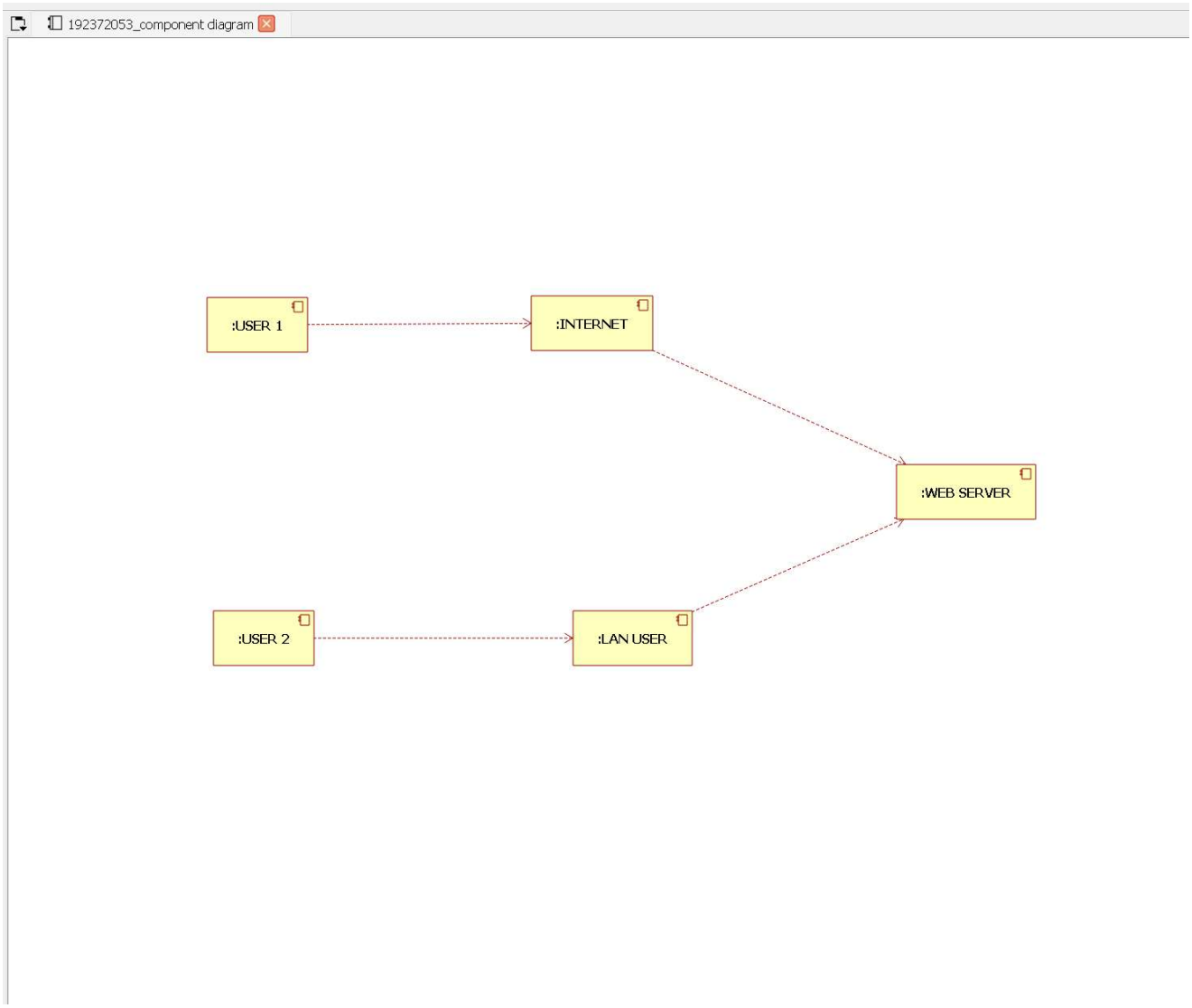


STATE CHART DIAGRAM:

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



REGISTER NO:

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.

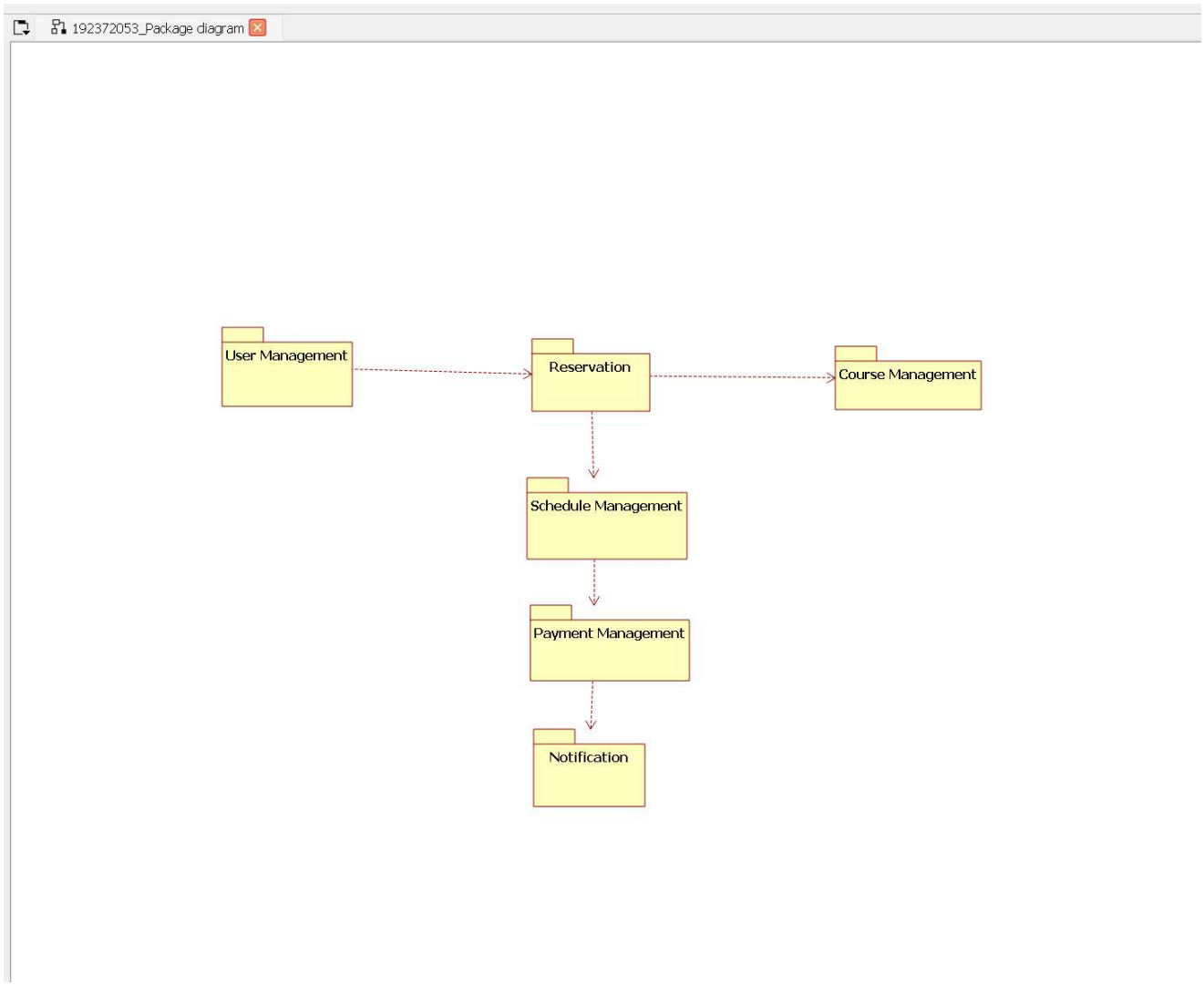
PACKAGE DIAGRAM:

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:

CENTRAL MANAGEMENT SYSTEM

```
Public class central management system
 {
        Public integer details;
        Public integer verify details;
        Public void verify()
         {
         }
        Public void enroll()
         {
```

```
        }
}
```

STUDENT:

```
Public class student
 {
```



```
        Public integer name;
        Public integer address;
        Public integer marks;
        Public void browse()
         {
         }
        Public void select()
         {
         }
        Public void register()
         {
         }
}
```

UNIVERSITY:

```
Public class university
 {
        Public integer store details;
        Public integer verify
        details; Public voidverify()
         {
         }
```

```
        Public void enroll()
        {
        }
}
```


RESULT:

Thus draw the diagrams [usecase, activity, sequence, collaboration, class,state chart, component, deployment , package] for the Online course reservation system has been designed executed and output is verified.

EX NO:6	E-TICKETING
DATE:	

AIM:

To draw the diagrams[use case, activity, sequence, collaboration, class, state chart, component, deployment, package] for the E-ticketing system.

SOFTWARE REQUIREMENTS SPECIFICATION

	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

In the E-Ticketing system the main process is a applicant have to login the database then the database verifies that particular username and password then the user must fill the details about their personal details then selecting the flight and the database books the ticket then send it to the applicant then searching the flight or else cancelling the process

1.3 PROJECT DESCRIPTION:

This software is designed for supporting the computerized e-ticketing. This is widely used by the passenger for reserving the tickets for their travel. This E-ticketing is organized by the central system. The information is provided from the railway reservation system.

1.4 REFERENCES:

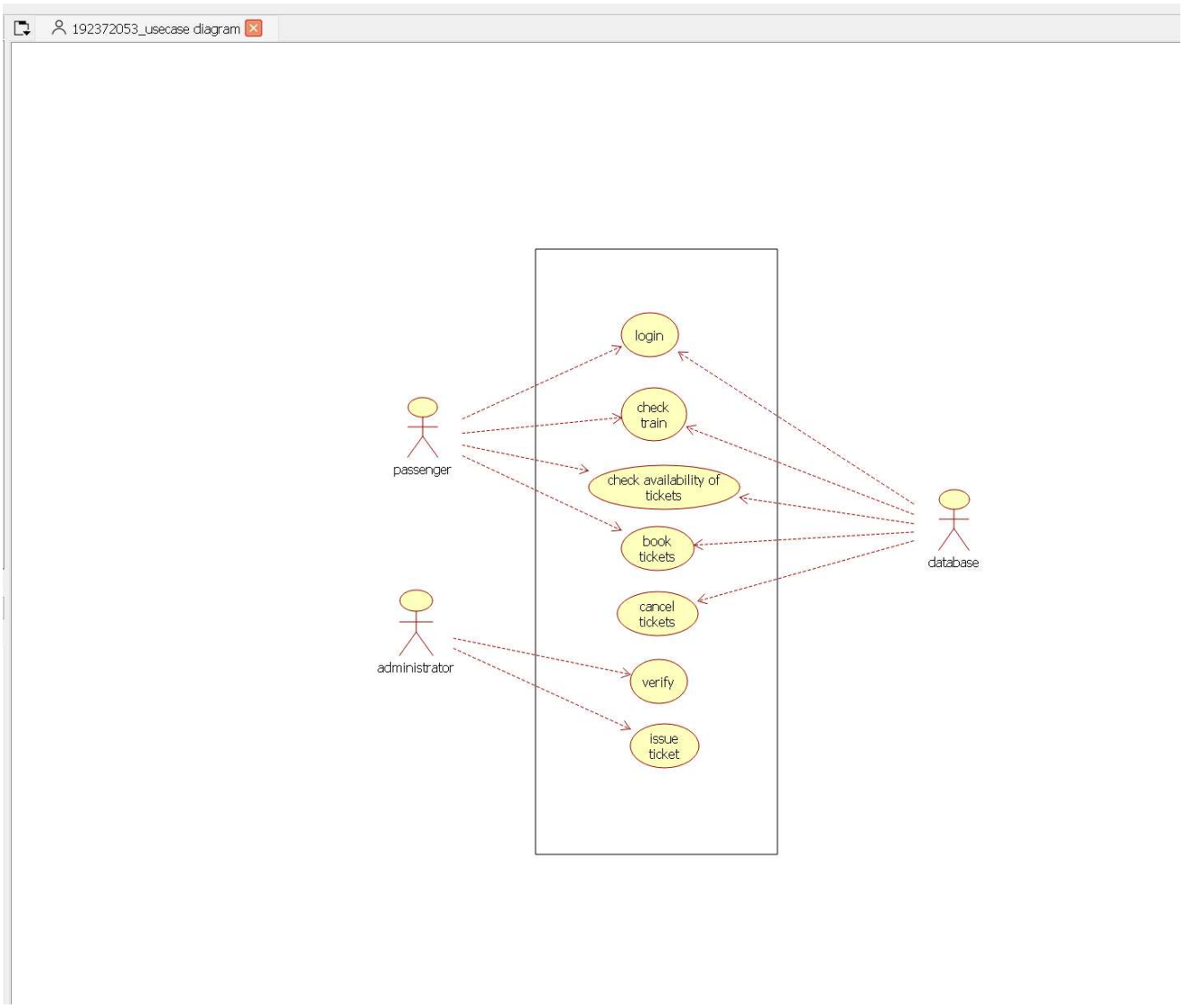
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Passenger, Railway reservation system..

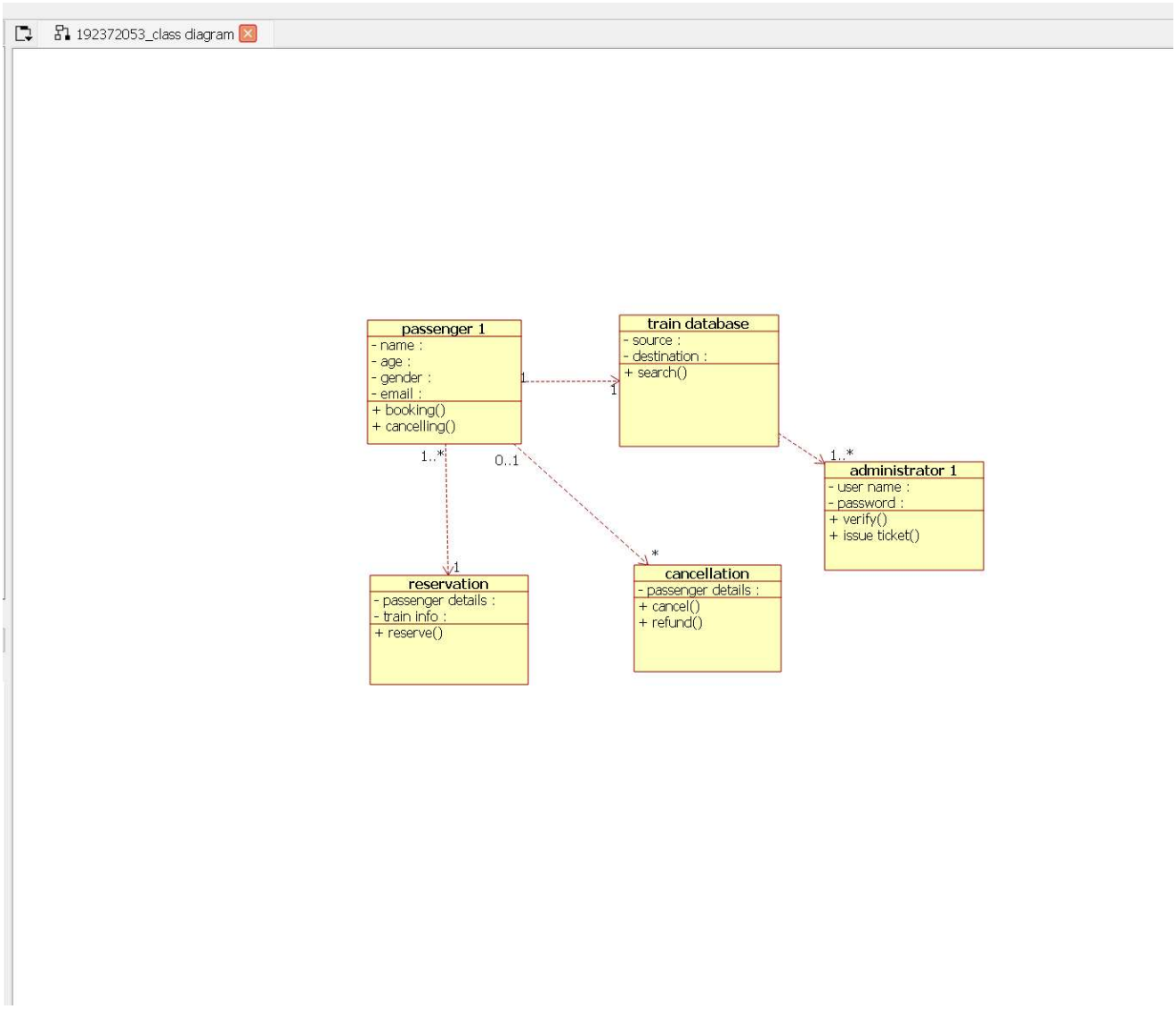
Use case: Status, reservation, cancellation, enter the train number, enter the number of seats, availability of seats, acceptance of ticket.



CLASS DIAGRAM:

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

CLASSES	ATTRIBUTES	OPERATIONS
Central computer	Train name, Passenger name	Reservation(), login()
Passenger	Passenger age	Login()
Railway reservation system	Train number	Cancellation()

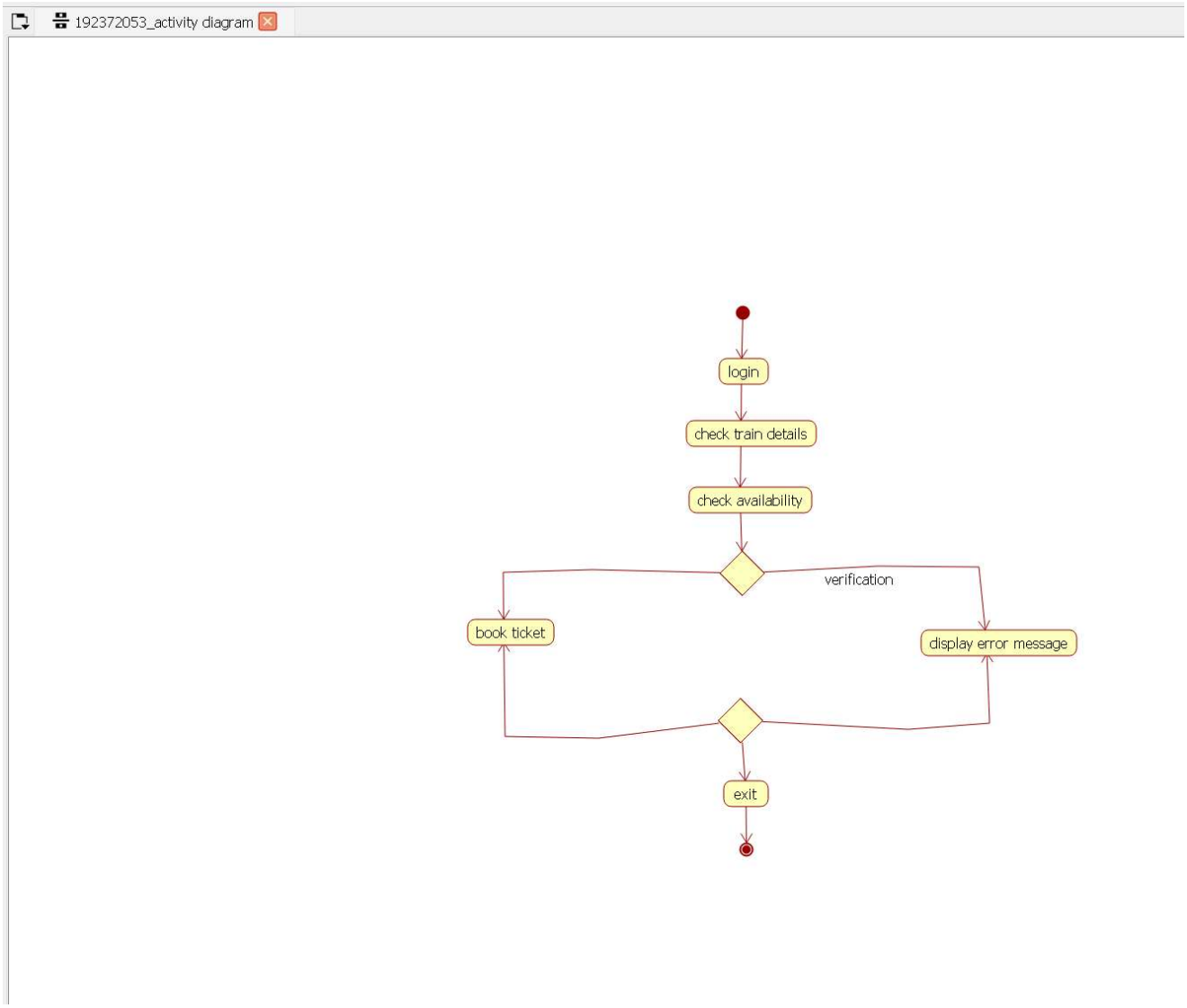


ACTIVITY DIAGRAM:

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

Activities: enter the train number, enter the number of seats, acceptance of ticket, accept seat.

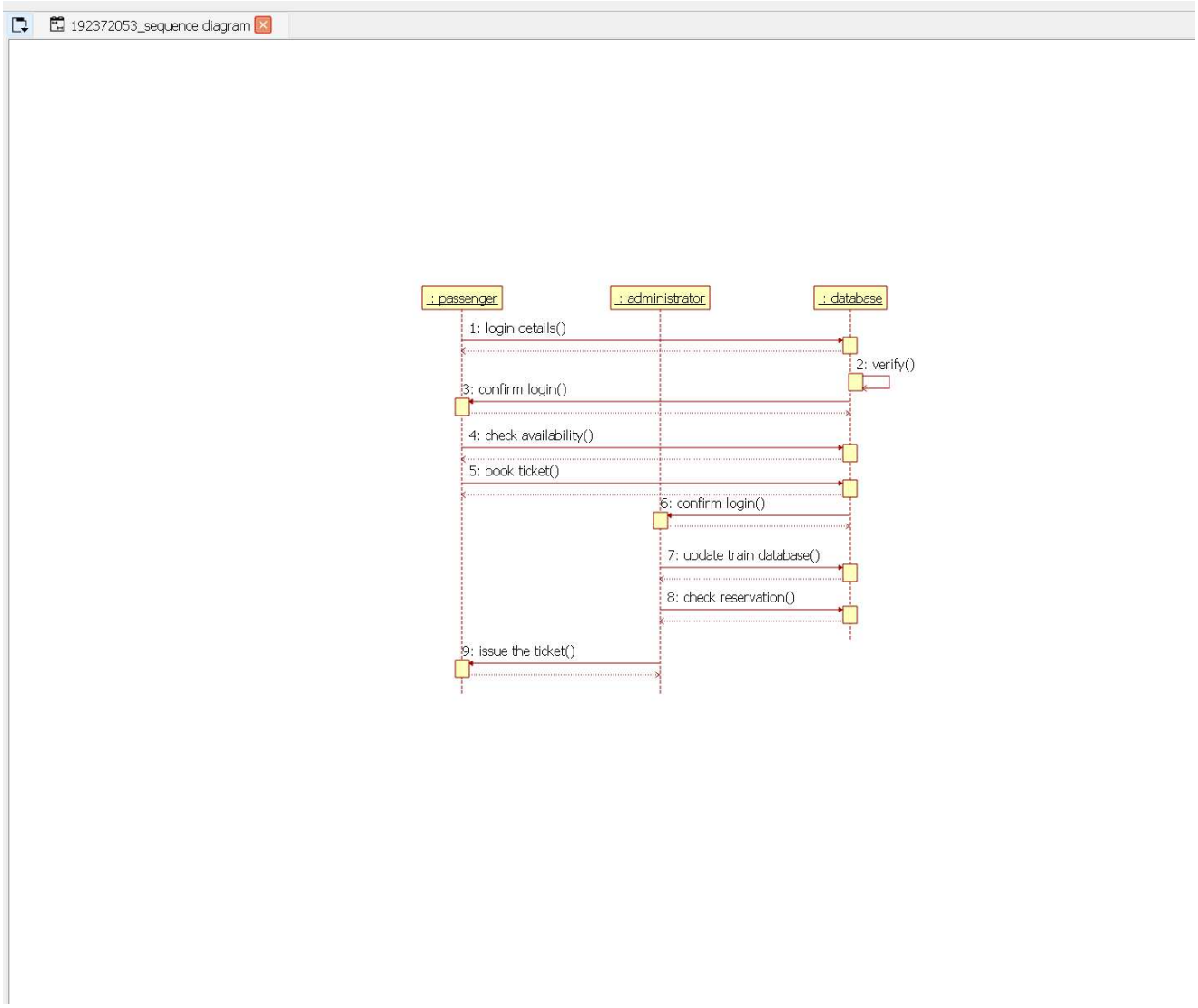
Decision box: Check availability of seats whether it is present or not.



SEQUENCE DIAGRAM:

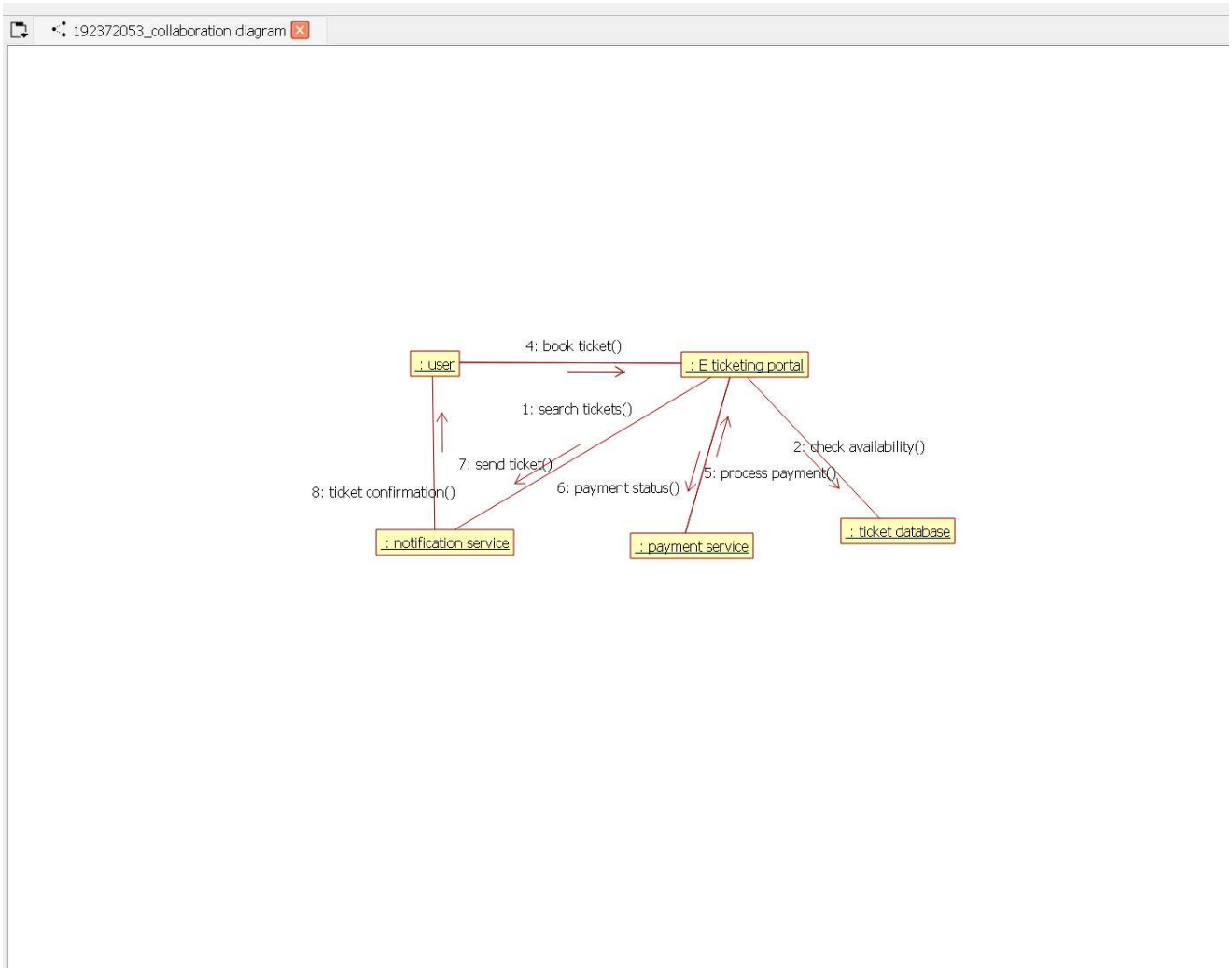
This diagram consists of the objects, messages and return messages.

Object: Passenger, Railway reservation system, Central computer.



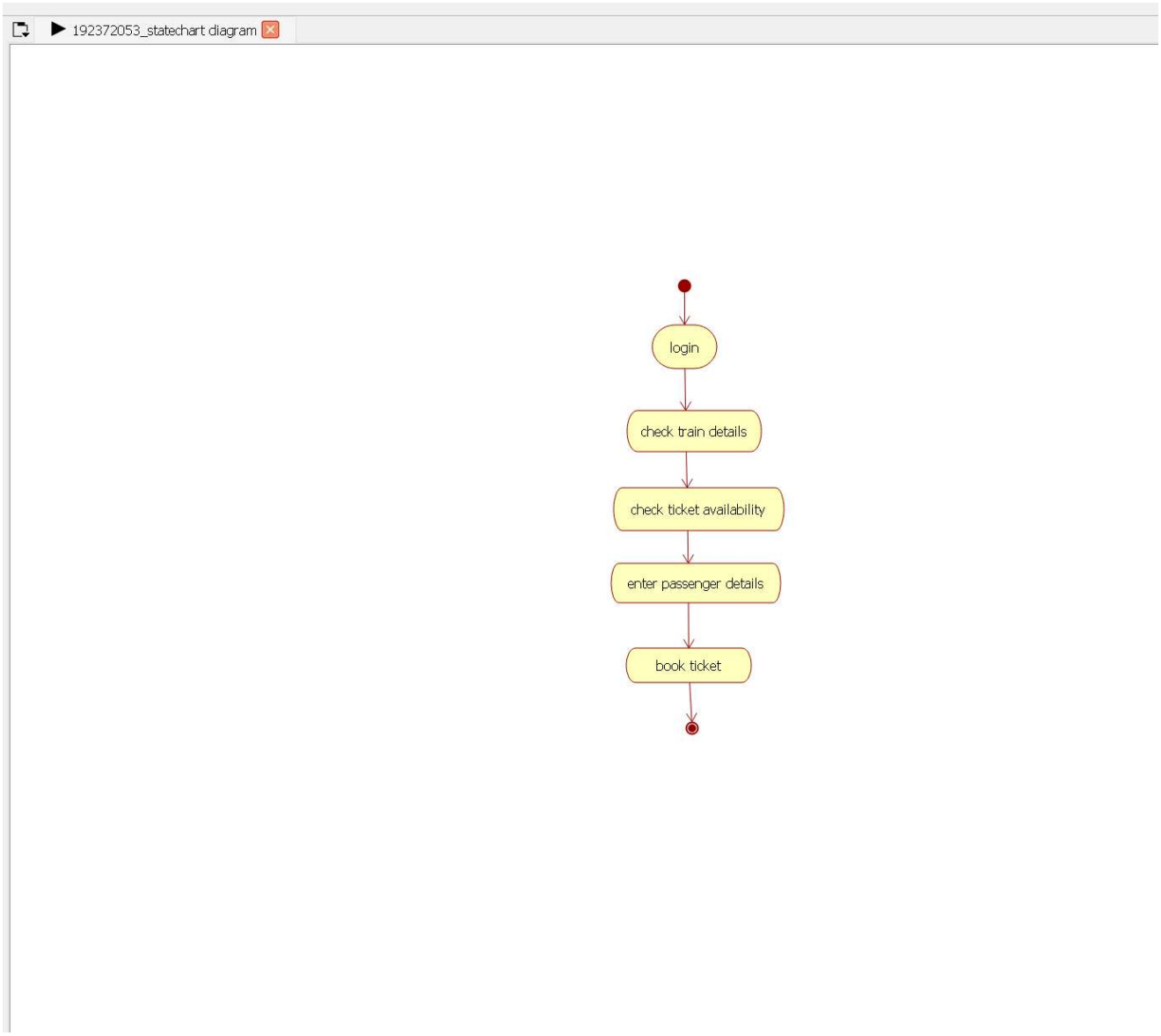
COLLABORATION DIAGRAM:

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:

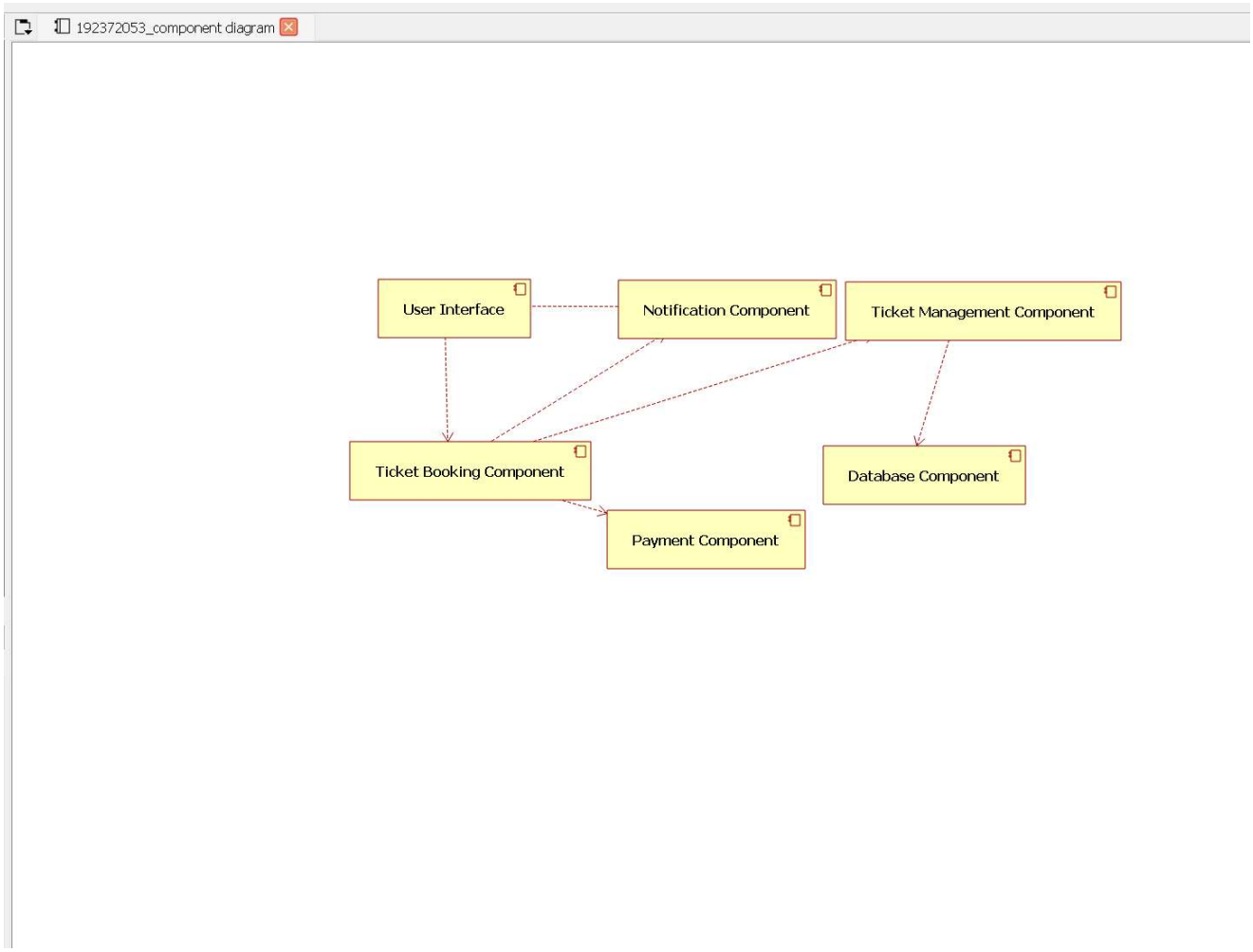
It is a technique to describe the behavior of the system. It describes all the possible states that a particular object gets into the object oriented technique. State diagram are drawn for a single class to show to the lifetime behaviour of a single objects



REGISTER NO:

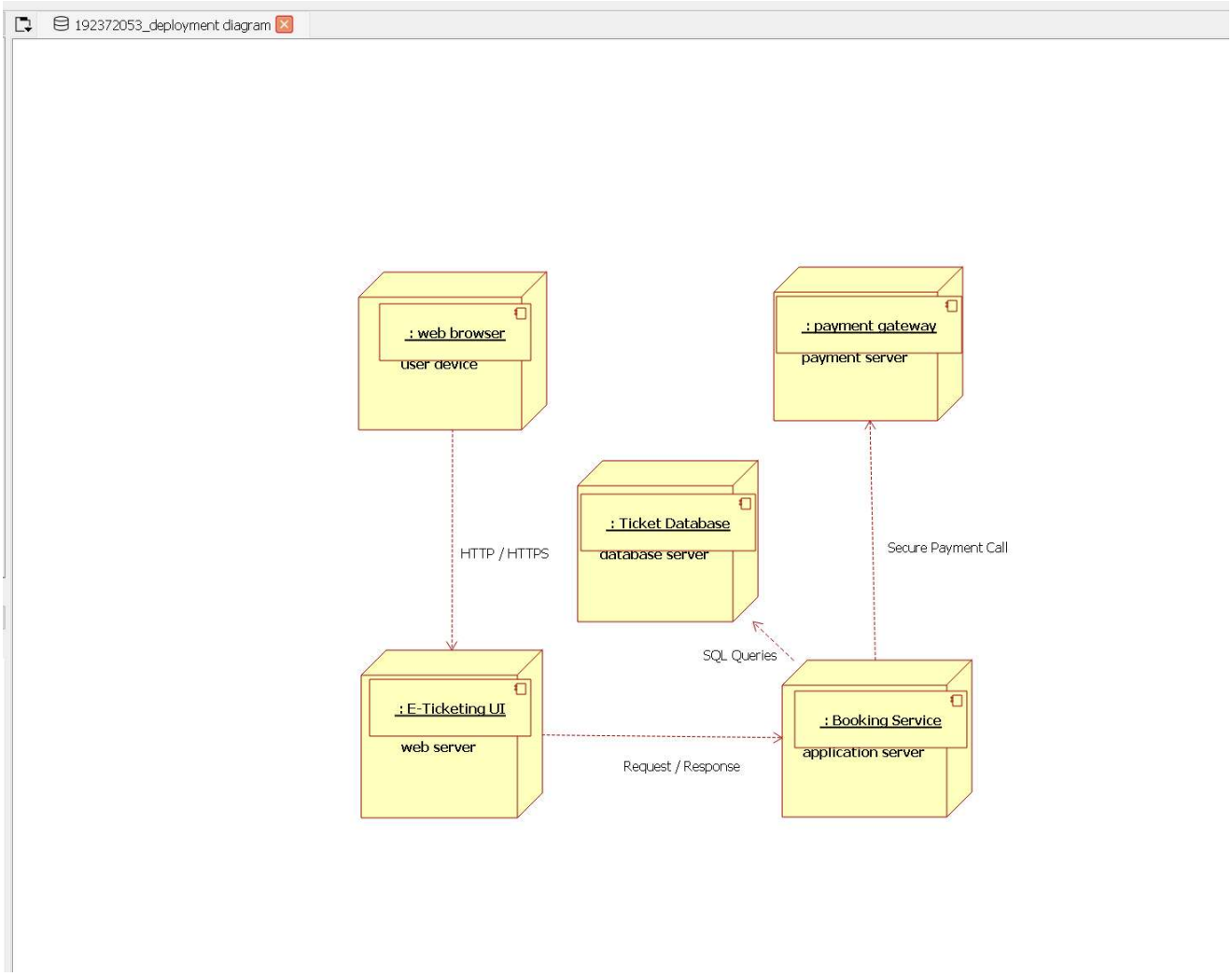
COMPONENT DIAGRAM:

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.

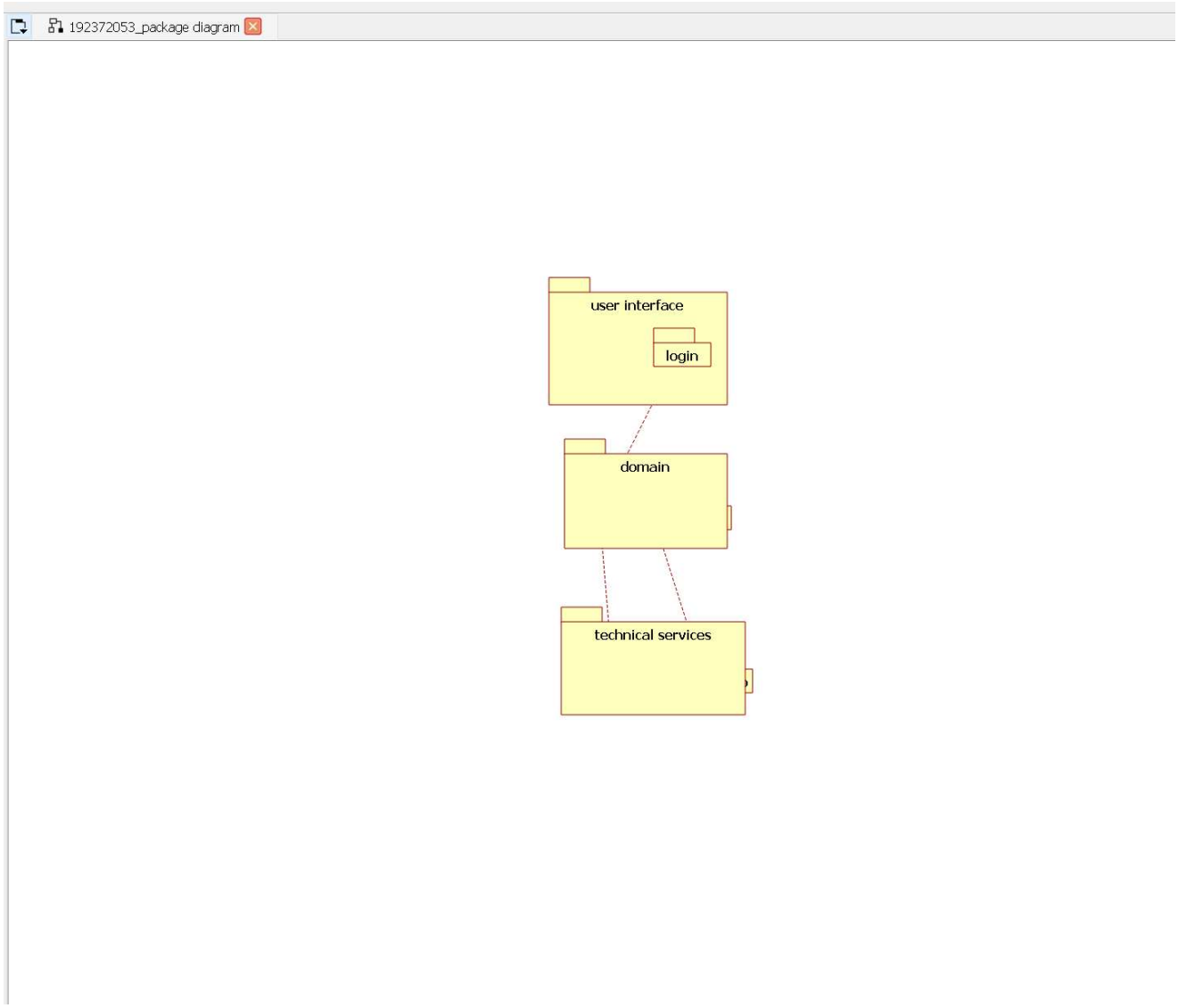


PACKAGE DIAGRAM:

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:

PASSENGER:

```
Public class passenger
 {
        Public integer passenger passenger name;
        Public integer passenger passenger age;
        Public integer train no;
        Public void passenger()
         {
         }
```

Public void new operation()

{
}

}

CENTRAL MANAGEMENT

SYSTEM: Public class central

management {

Public integer train name;

Public integer passenger name;

Public void reservation()

{

}

Public void cancellation()

{

}

Public void status()

{

}

Public void login()

{

}

Private void management()

{

}

}

RAILWAY RESERVATION SYSTEM:

Public class railway reservation system {

Public integer trainno;
Public integer train name;

Public integer passenger name;

Public void status()

{

}

Public void reservation()

{

}

Public void cancellation()

{

}

Public void railway reservation system()

{

}

}

RESULT:

Thus the diagrams[use case, activity, sequence, collaboration, class, statechart, component, deployment, package] for the E-ticketing system has been designed, executed and output is verified.

EX NO:07	CREDIT CARD PROCESSING SYSTEM
DATE:	

AIM:

To draw the diagrams [usecase, activity, sequence, collaboration, class, statechart,

component, deployment, package] for Credit Card Processing .

SOFTWARE REQUIREMENTS SPECIFICATION:

	SOFTWARE REQUIREMENTS SPECIFICATION
1.0	Hardware Requirements
1.1	Software Requirements

REGISTER NO:

1.3	Project description
1.4	Reference

OOAD LAB

1.2	Problem Analysis and l

1.0 HARDWARE REQUIREMENTS:

Intel Pentium Processor I3/I5

1.1 SOFTWARE REQUIREMENTS:

Rational rose / Argo UML

1.2 PROBLEM ANALYSIS AND PROJECT PLANNING

The Credit Card Processing System which is use to purchasing an item from any shop mall, and it is used to maintain the limitation of credit card balance and current transaction process could be update via credit card machine. This project mainly used for large amount of item can be easy to buy from anywhere and required transaction process should be maintained them.

1.3 PROJECT DESCRIPTION:

This software is designed for supporting the computerized credit card processing System .In this system, the cardholder purchases items and pays bill with the aid of the credit card. The cashier accepts the card and proceeds for transaction using the central system. The bill is verified and the items are delivered to the cardholder.

1.4 REFERENCES:

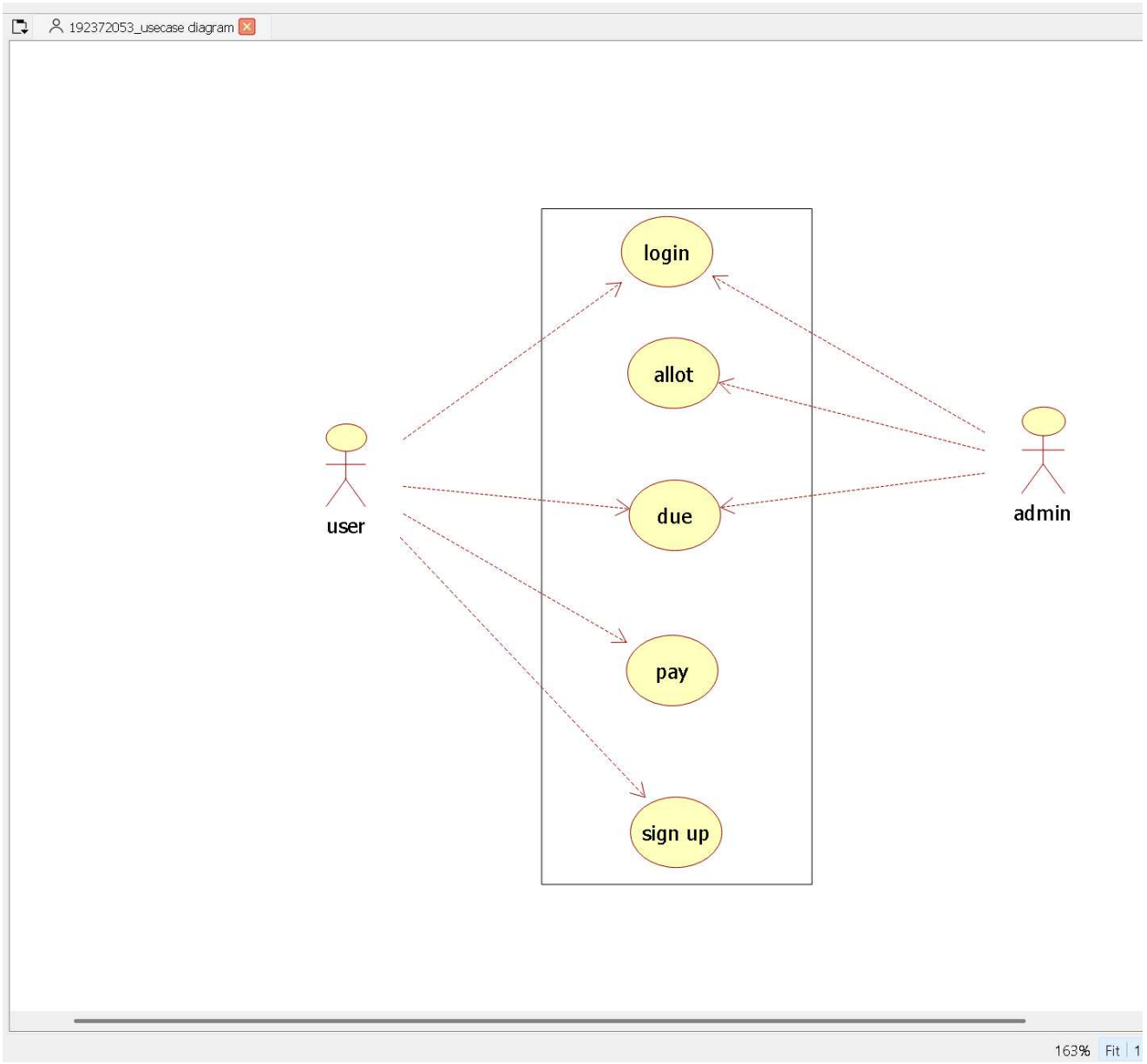
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Cardholder, Cashier, Central system.

Use case: Receive bill, Give card, Enter card number, Enter amount, Transaction, Receive Receipt.

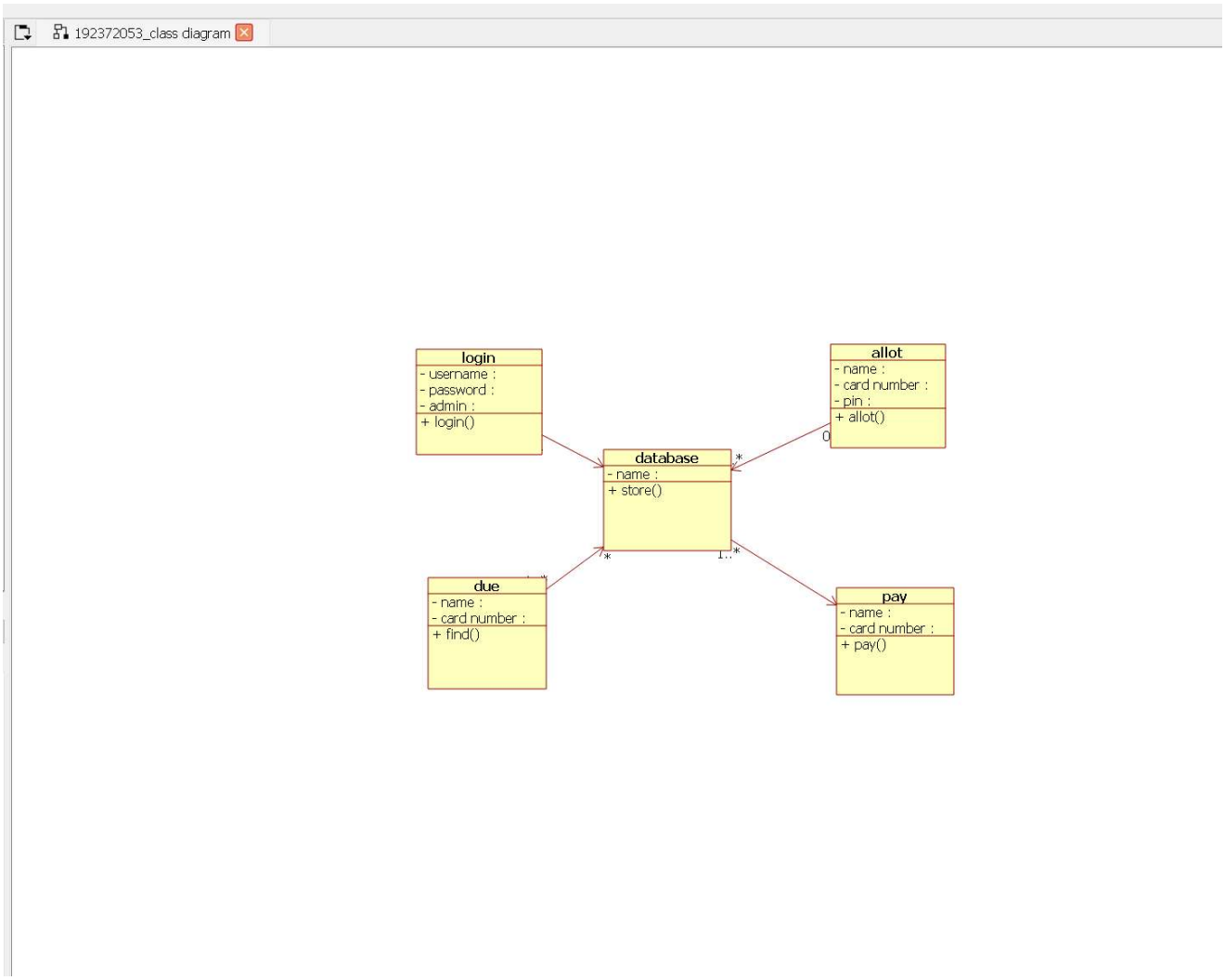


CLASS DIAGRAM:

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

CLASSES	ATTRIBUTES	OPERATIONS
Central system	product name, product details	Print bill(), Validate card()
Cashier	Product name, Cost of the product	Enter amount(), Swipe card(), Print bill(), Deliver product()

Card holder	Item purchased, Validate card	Give card(), Sign bill()



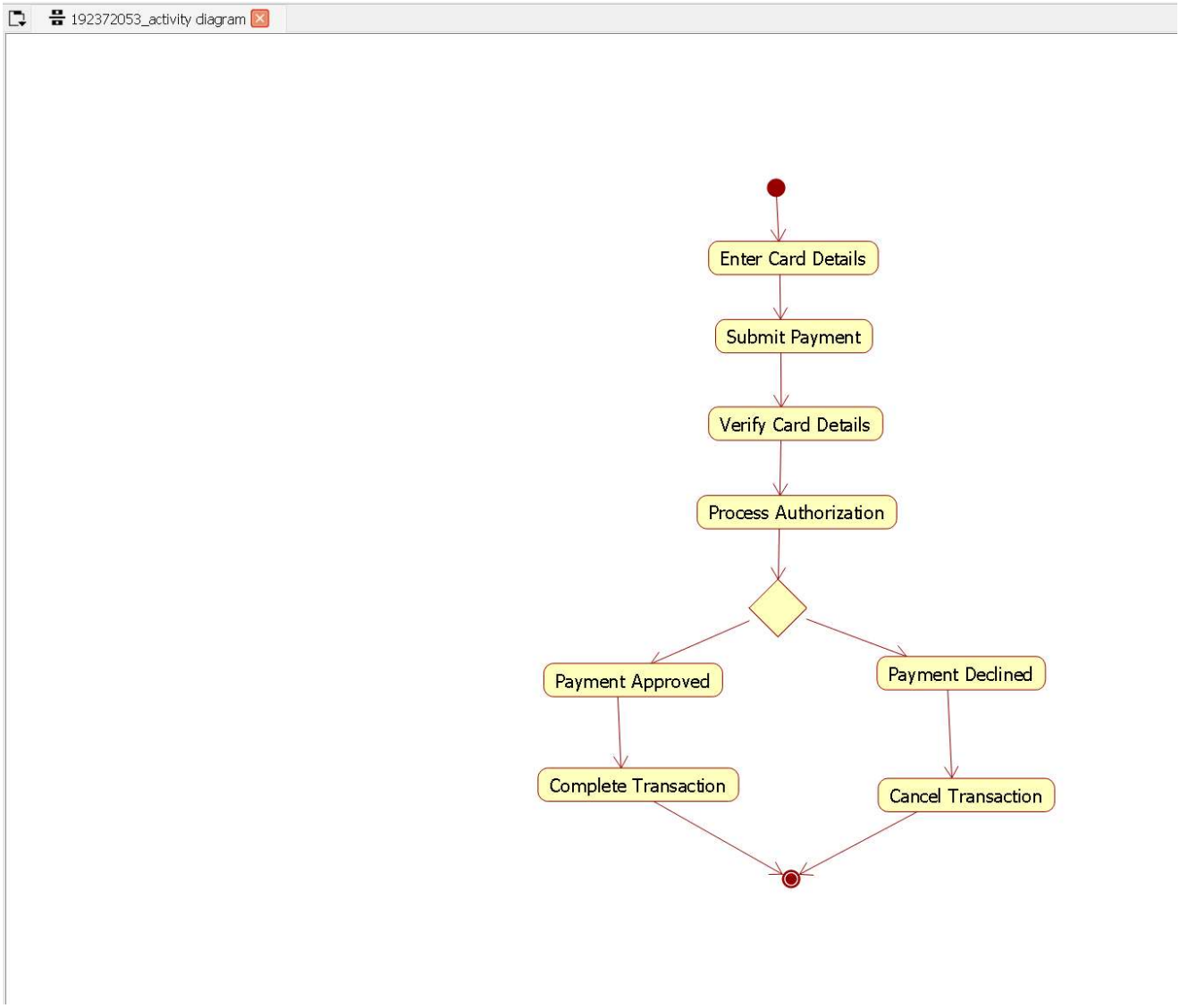
REGISTER NO:

ACTIVITY DIAGRAM:

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

Activities: Receive Bill, Give card, Enter the card number, Enter the amount, Transaction, Receive Receipt

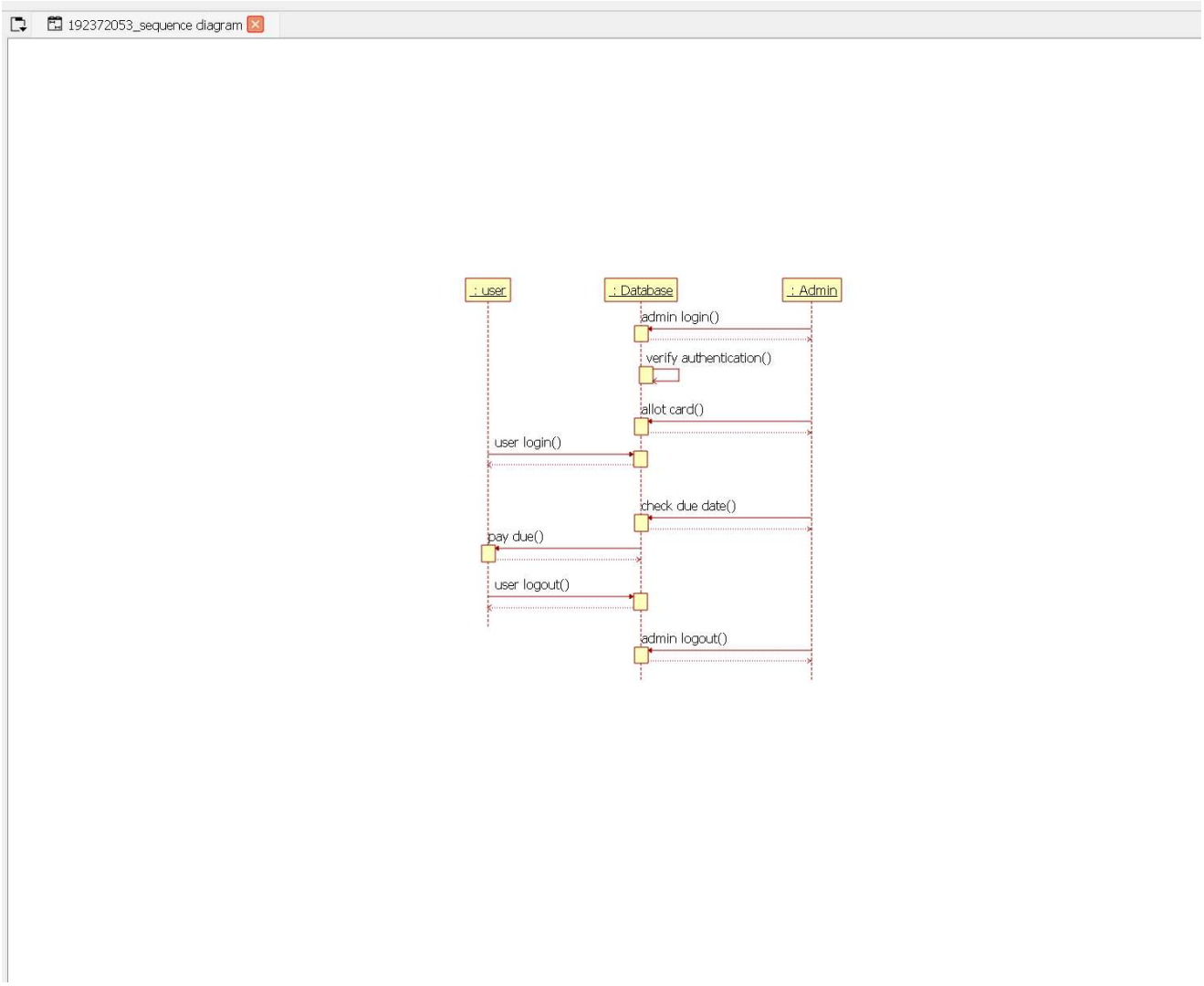
Decision box: Verification of card



SEQUENCE DIAGRAM:

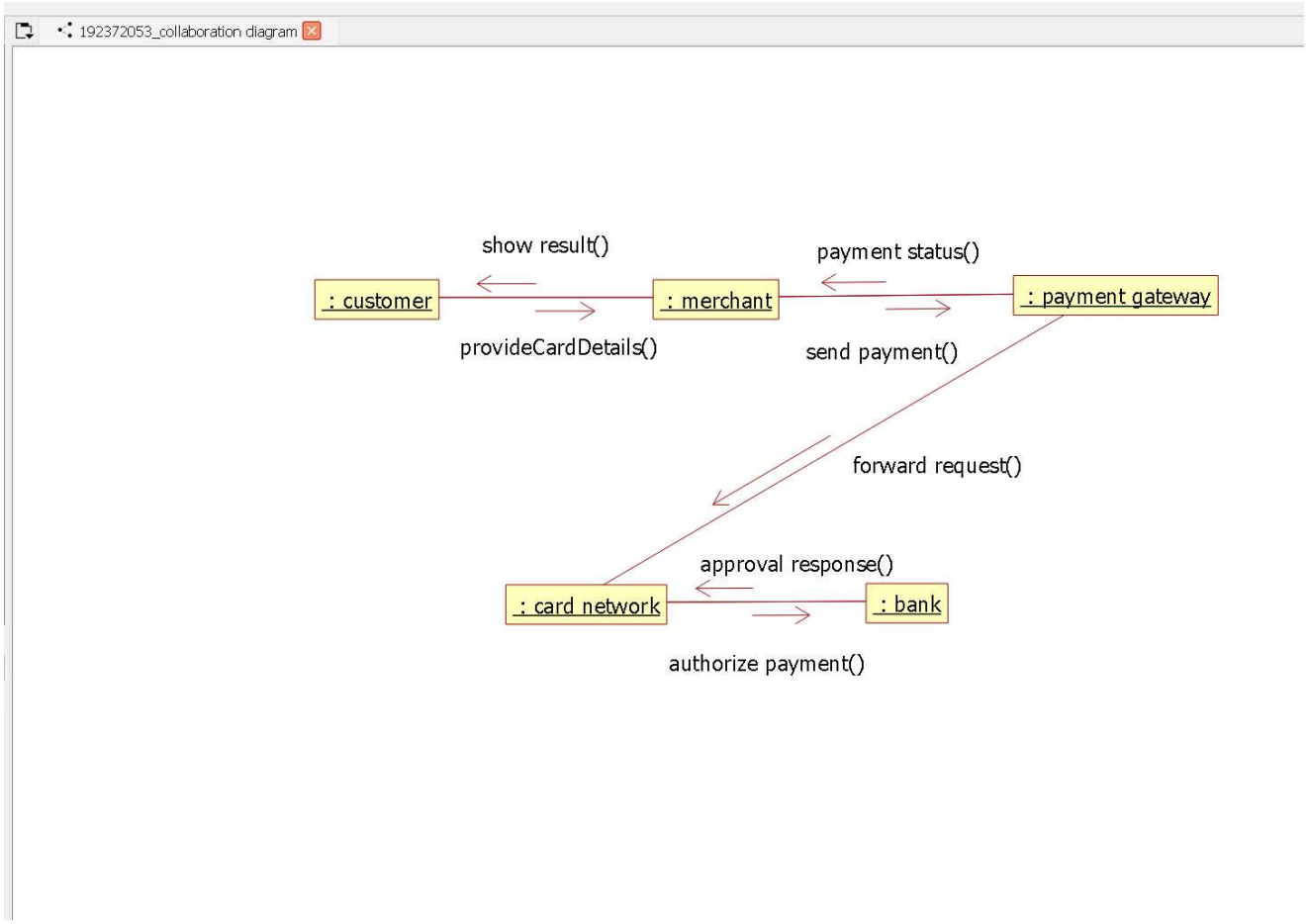
This diagram consists of the objects, messages and return messages.

Object: Card Holder,Cashier , Central system



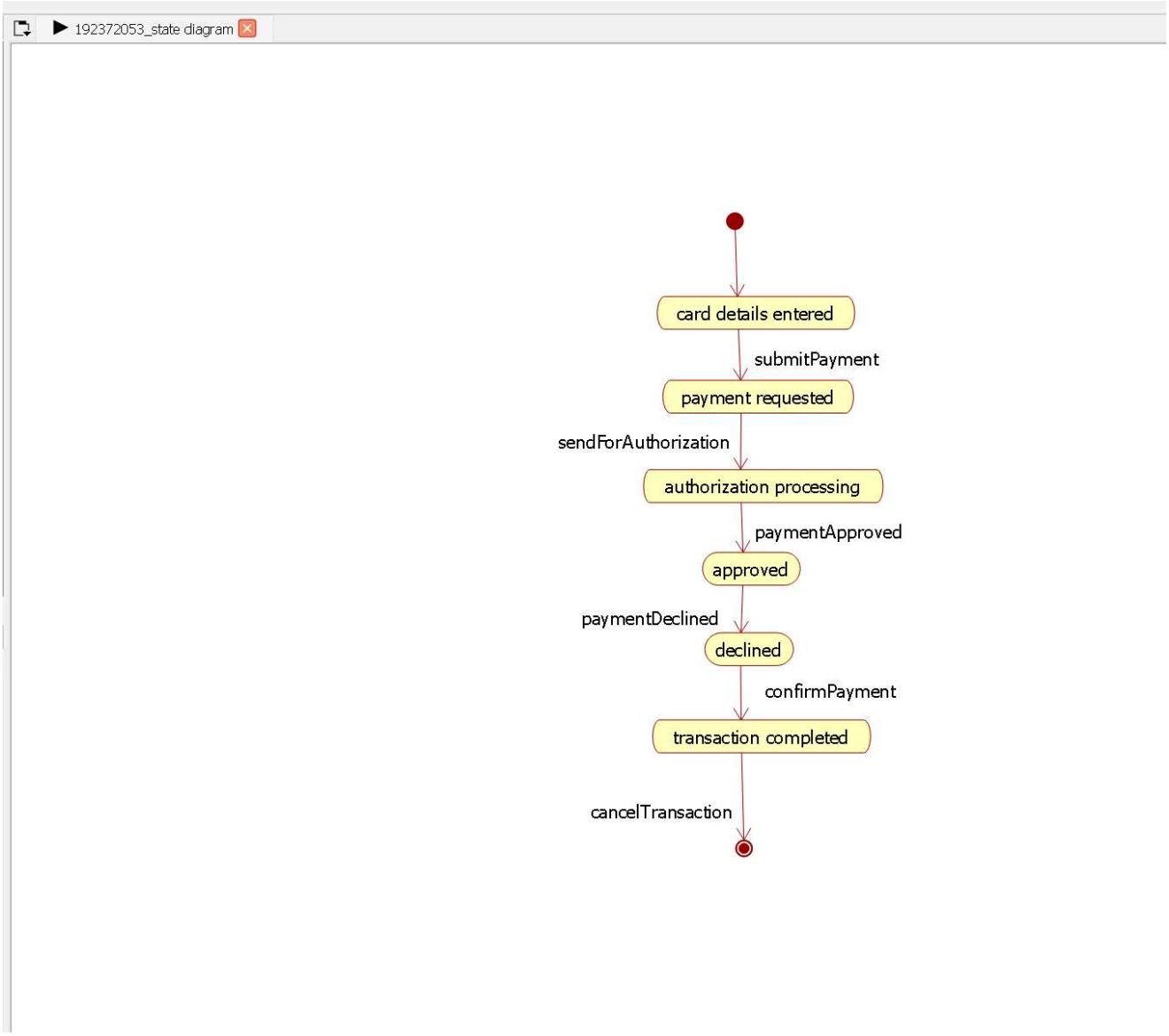
COLLABORATION DIAGRAM:

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



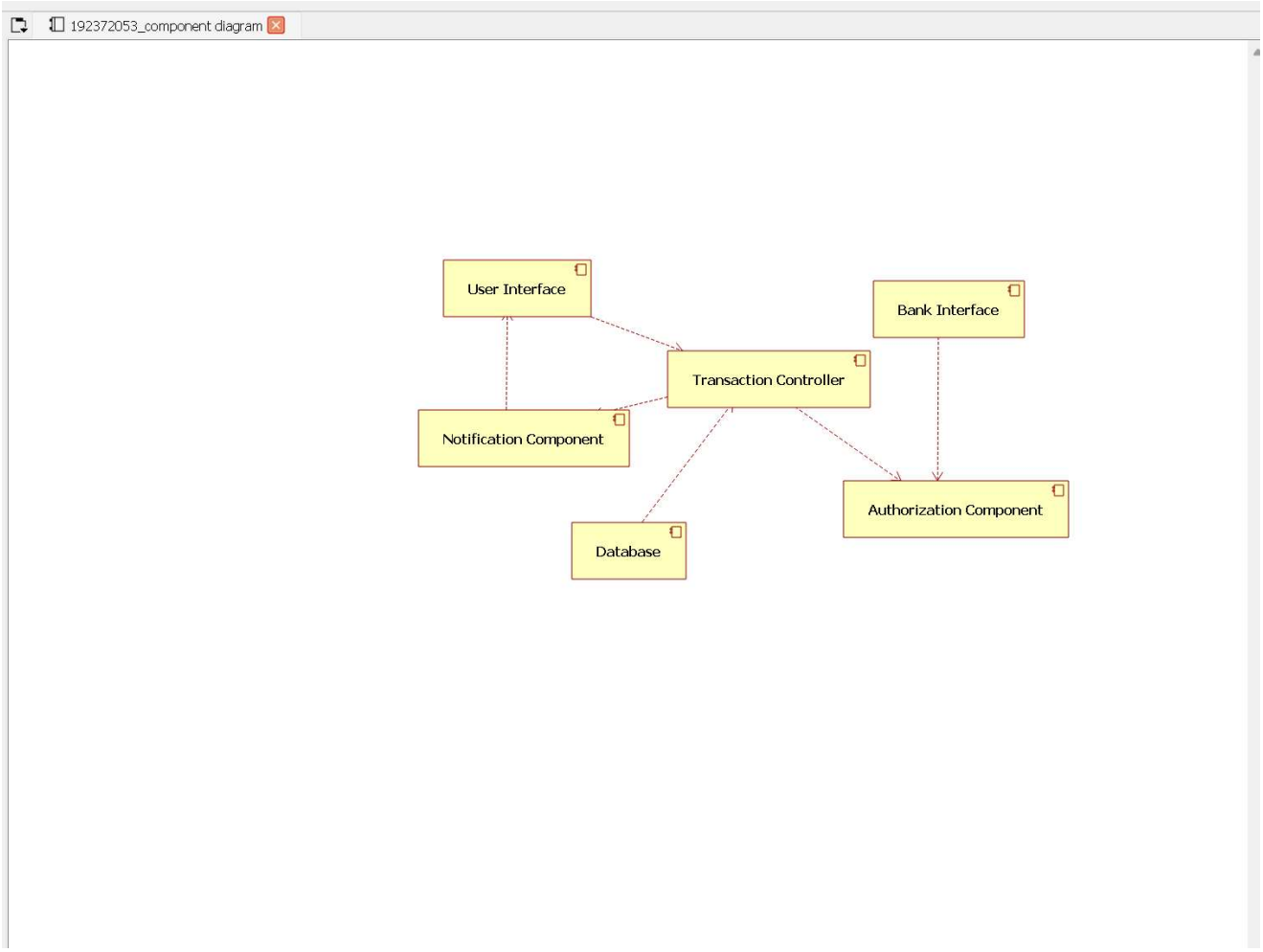
STATE CHART DIAGRAM:

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



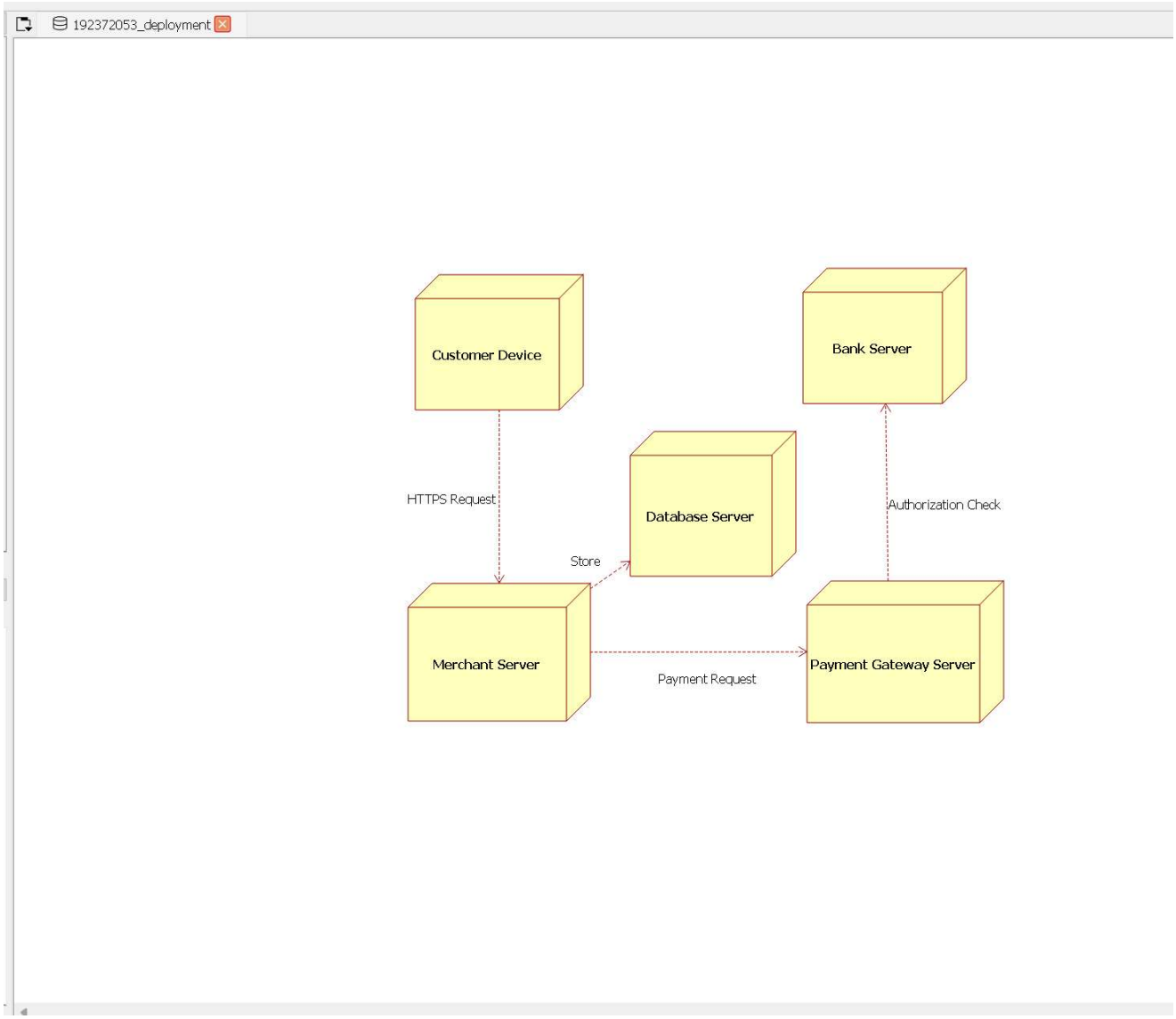
COMPONENT DIAGRAM:

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.

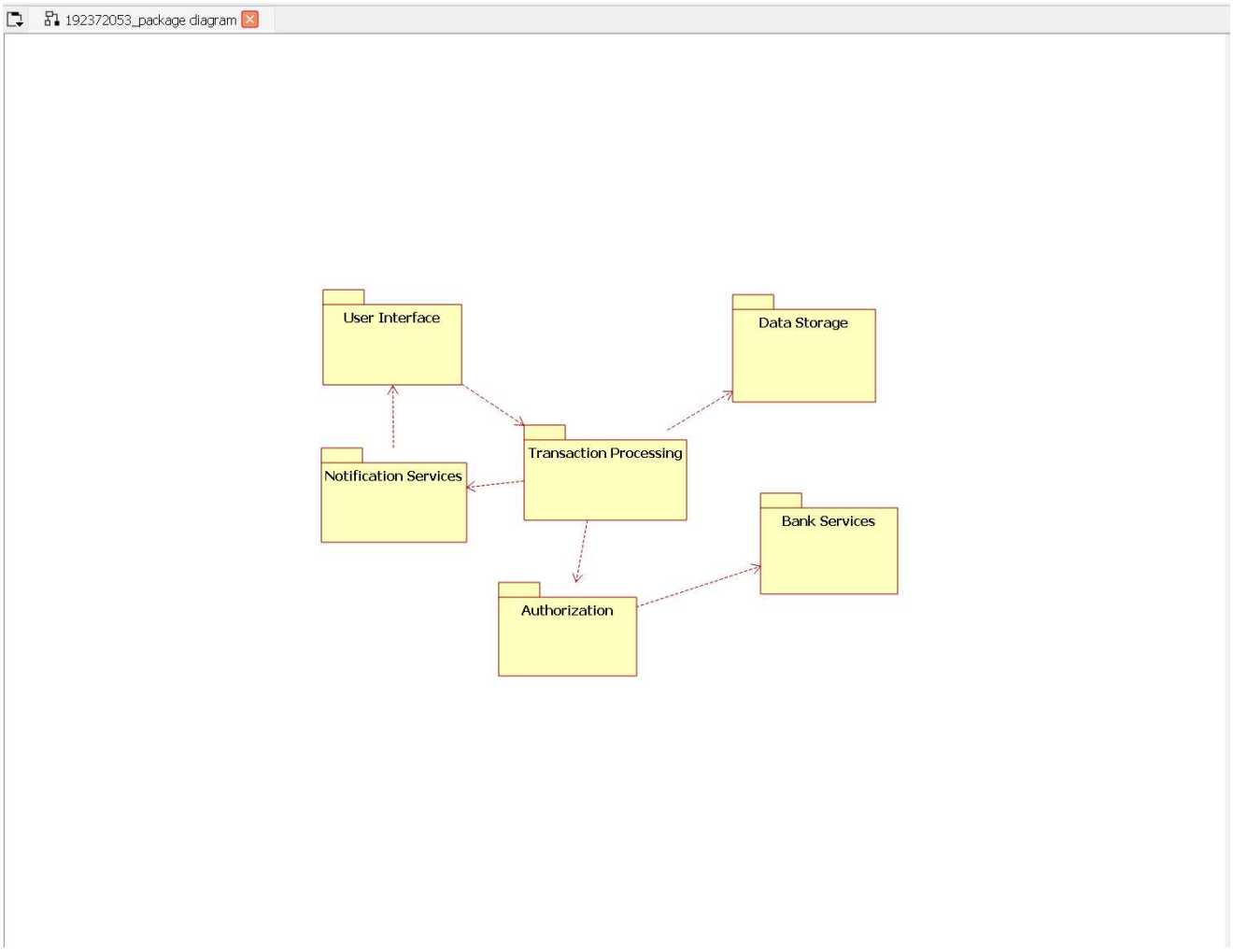


PACKAGE DIAGRAM:

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



REGISTER NO:

o Technical services layer

PROGRAM CODING:

CASH HOLDER:

```
public class cash holder
        {

    public Integer itemspurchased;

     public void signbill()
            {
            }

    }
```

CASHIER:

```
public class cashier
  {
```

```
public Integer name;

public Integer cast;

public void amount()
        {
        }

}
```

CENTRAL SYSTEM:

```
public class central sys
 {

        private Integer productname;
        public Integer productdetails;

        public void printbill()
         {
         }
        public void validatecard()
         {
         }
```


RESULT:

Thus the diagrams[use case, activity, sequence, collaboration, class, state chart, component, deployment, package] for the credit card processing system has been designed, executed and output is verified.

EX:NO:08	**SOFTWARE PERSONNEL MANAGEMENT SYSTEM**
DATE:	

AIM:

To draw the diagrams [Usecase, Class, Activity, Sequence, Collaboration, State chart, Deployment, Component, package] for software personnel management system.

SOFTWARE REQUIREMENTS SPECIFICATION:

	SOFTWARE REQUIREMENTS SPECIFICATION
1.0	Hardware Requirements
1.1	Software Requirements
1.2	Project description
1.3	Reference

1.0 HARDWARE REQUIREMENTS:

Intel Pentium Processor I3/I5

1.1 SOFTWARE REQUIREMENTS:

Rational Rose/Argo UML

1.2 PROJECT DESCRIPTION:

This software is designed for the process of knowing the details of a person works in a software company. The details are being stored in the Central Management System for the cross checking the person's details.

1.3 REFERENCES:

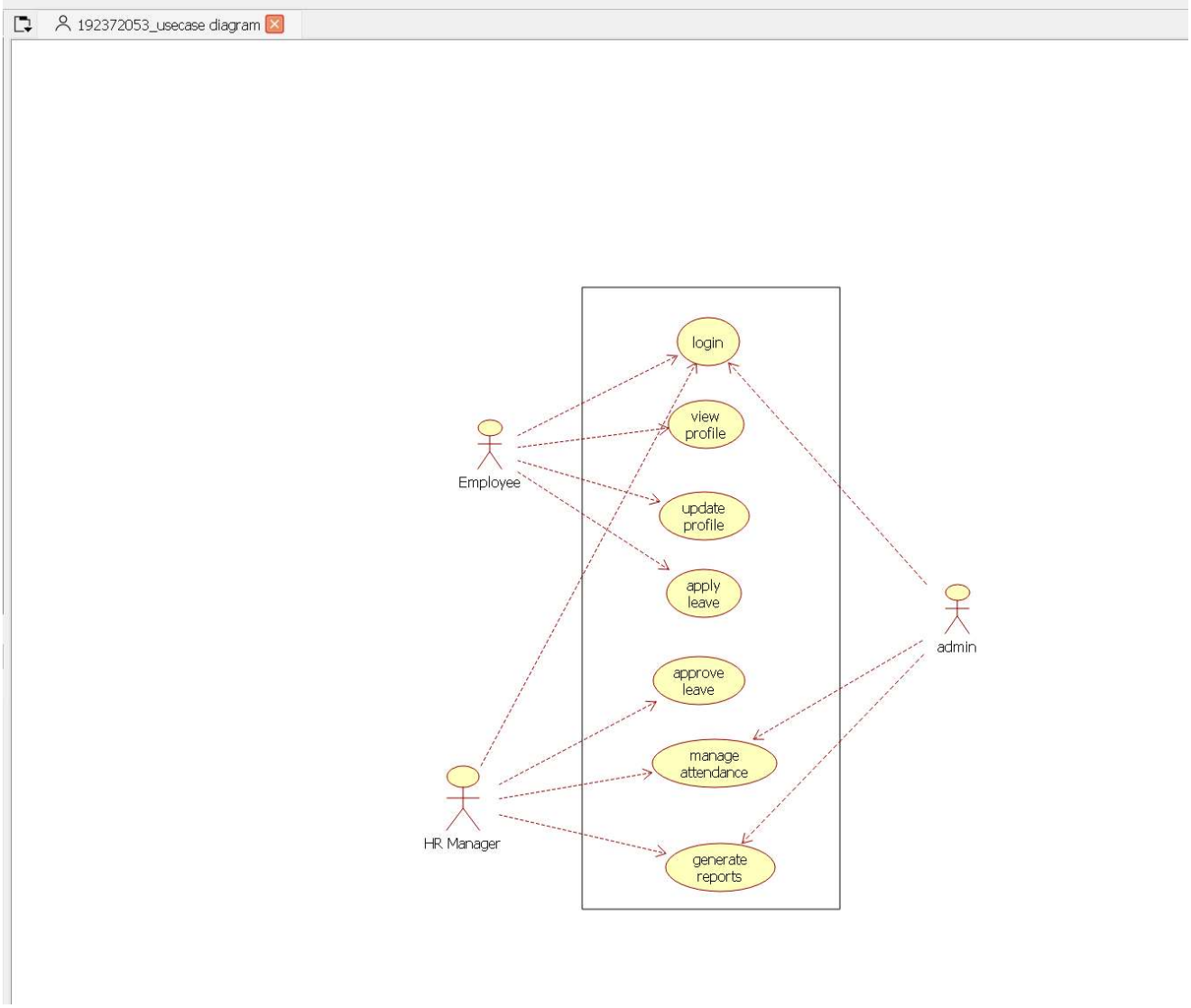
IEEE Software Requirement Specification format.

USECASE DIAGRAM:

This diagram will contain the actors, usecases which are:

ACTORS: Employee, HR, Central System

USECASE: Name and address, qualification, experience, internet, loan, verificatio

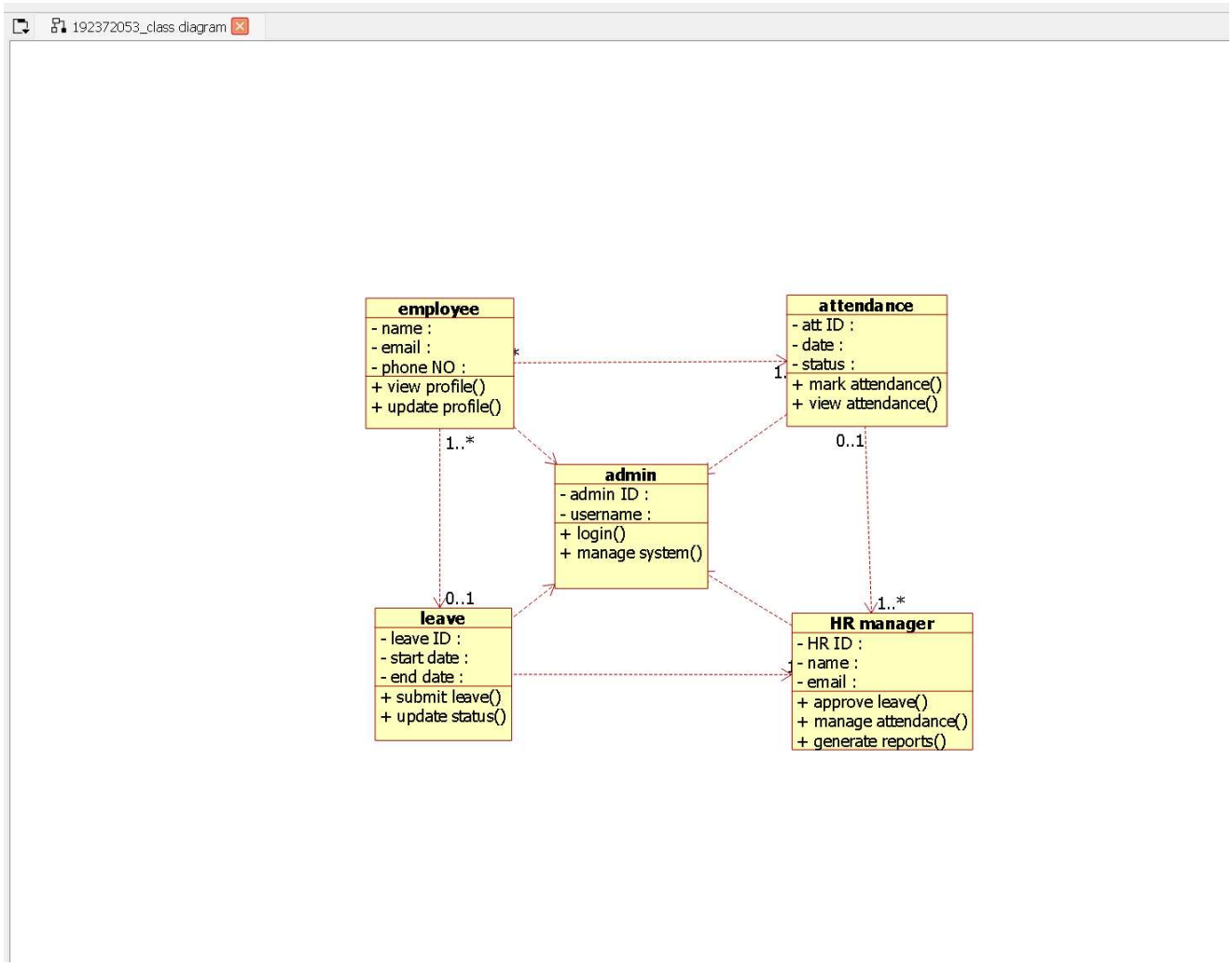


REGISTER NO:

CLASS DIAGRAM:

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

CLASSES	ATTRIBUTES	OPERATIONS
Central Management System	Employee name Employee number	Tax() Loan()
Employee1	Employee details	Leave taken()
HR	Check details	Loss of pay()

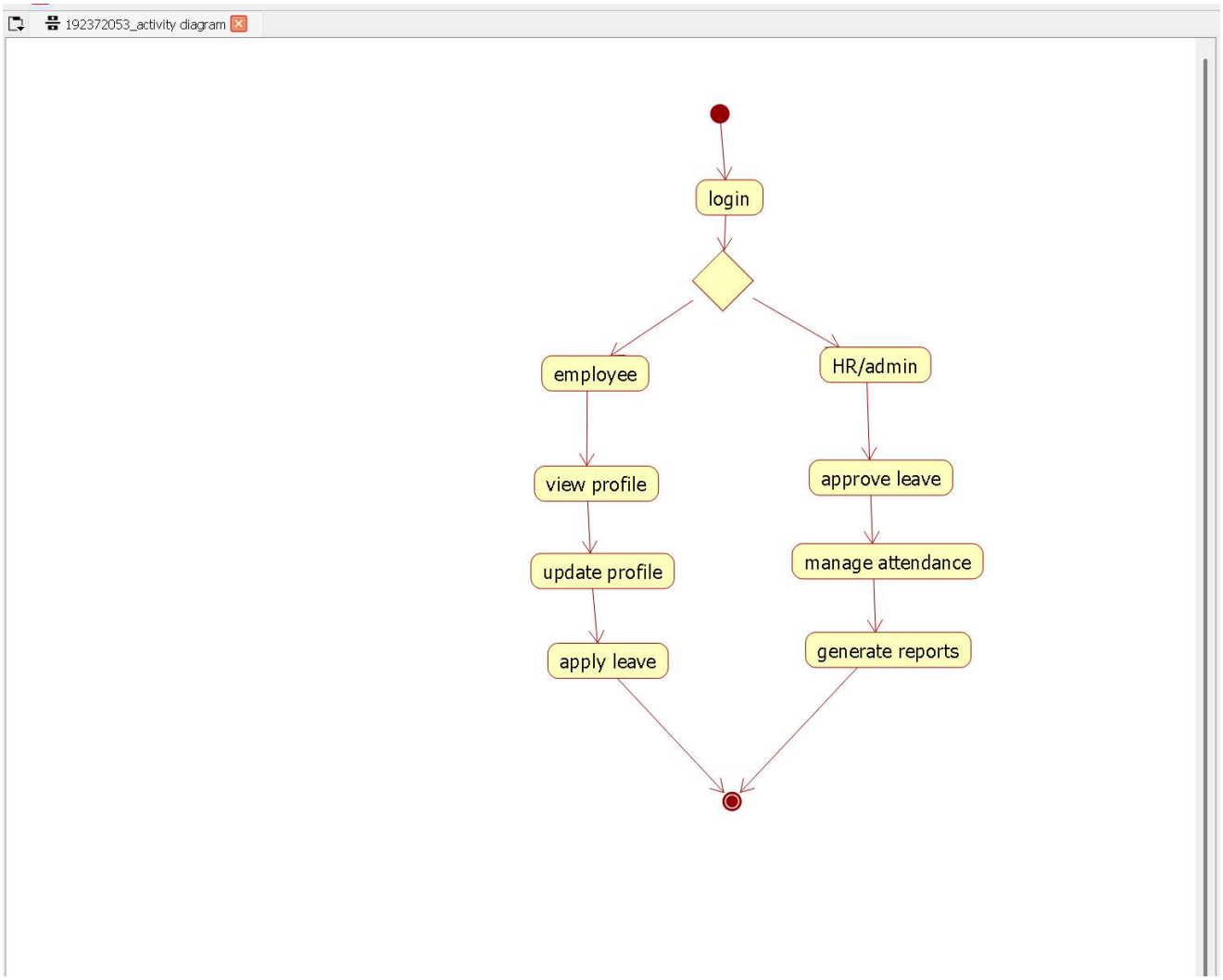


ACTIVITY DIAGRAM:

This diagram will have the activities as start point, end point, decision boxes as:

ACTIVITIES: Enter the option to check, enter the salary, enter the working days, leave taken, loss of pay

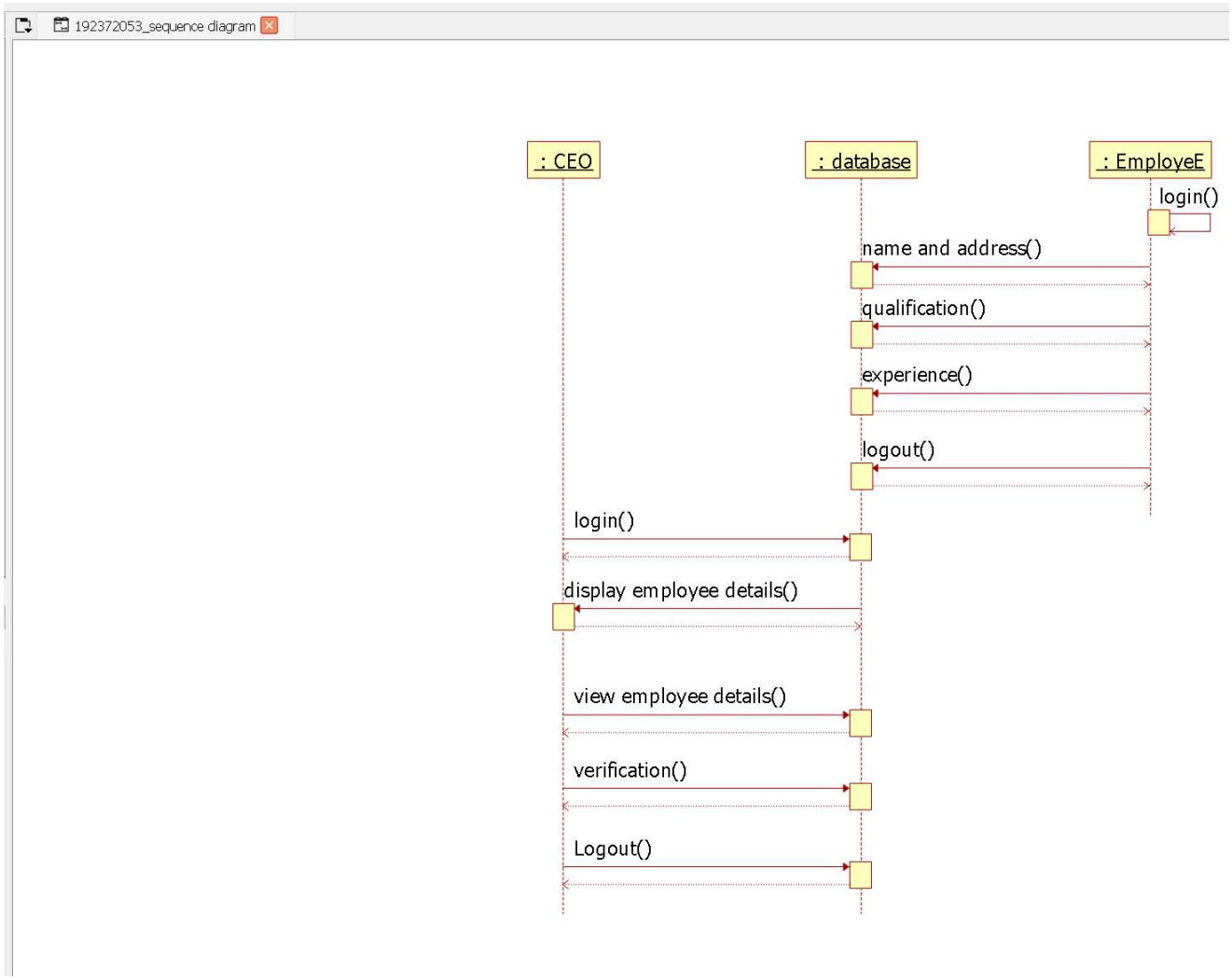
DECISION BOX: Option to check



SEQUENCE DIAGRAM:

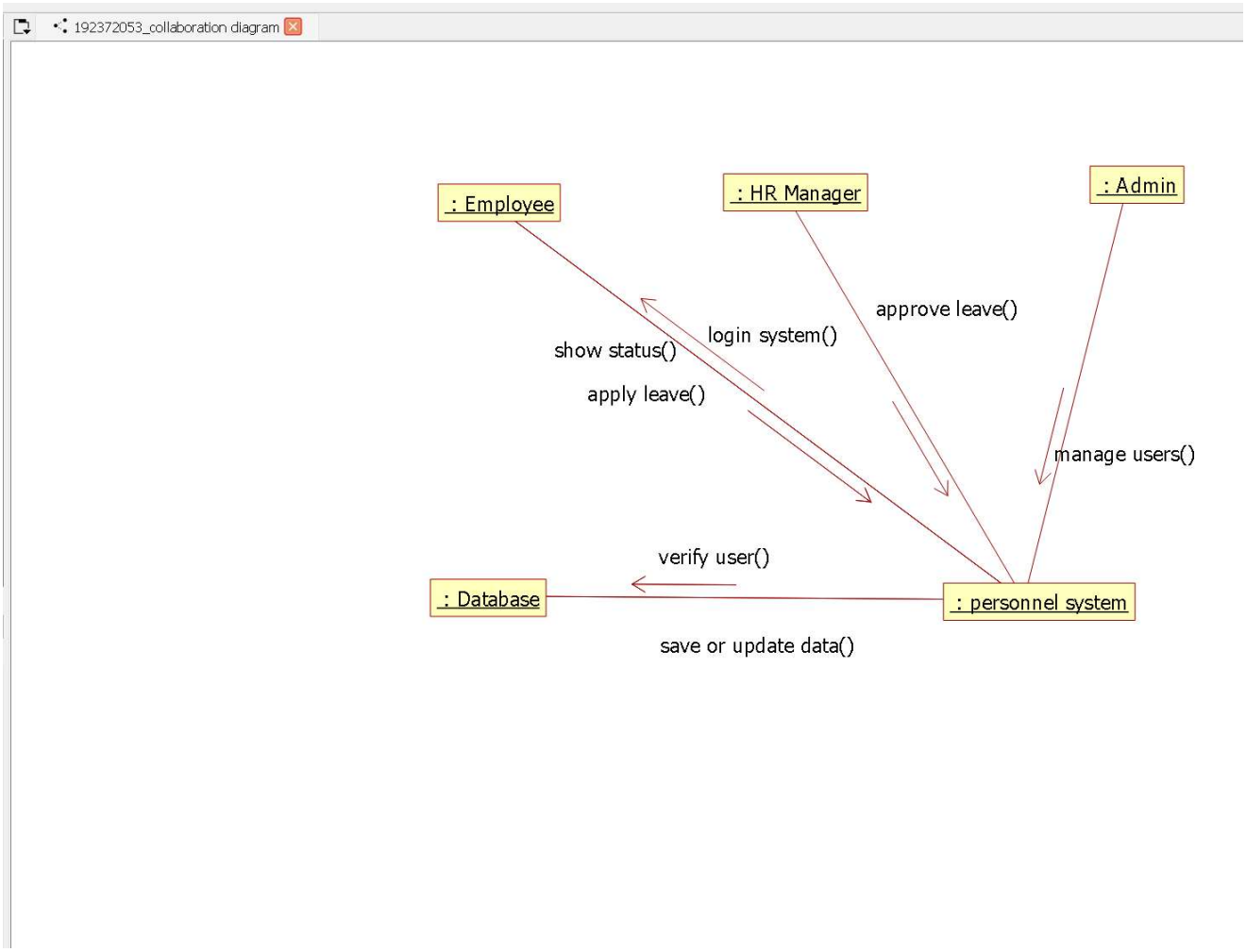
This diagram consists of the objects, messages and return messages

OBJECT: Employee, HR, Central System



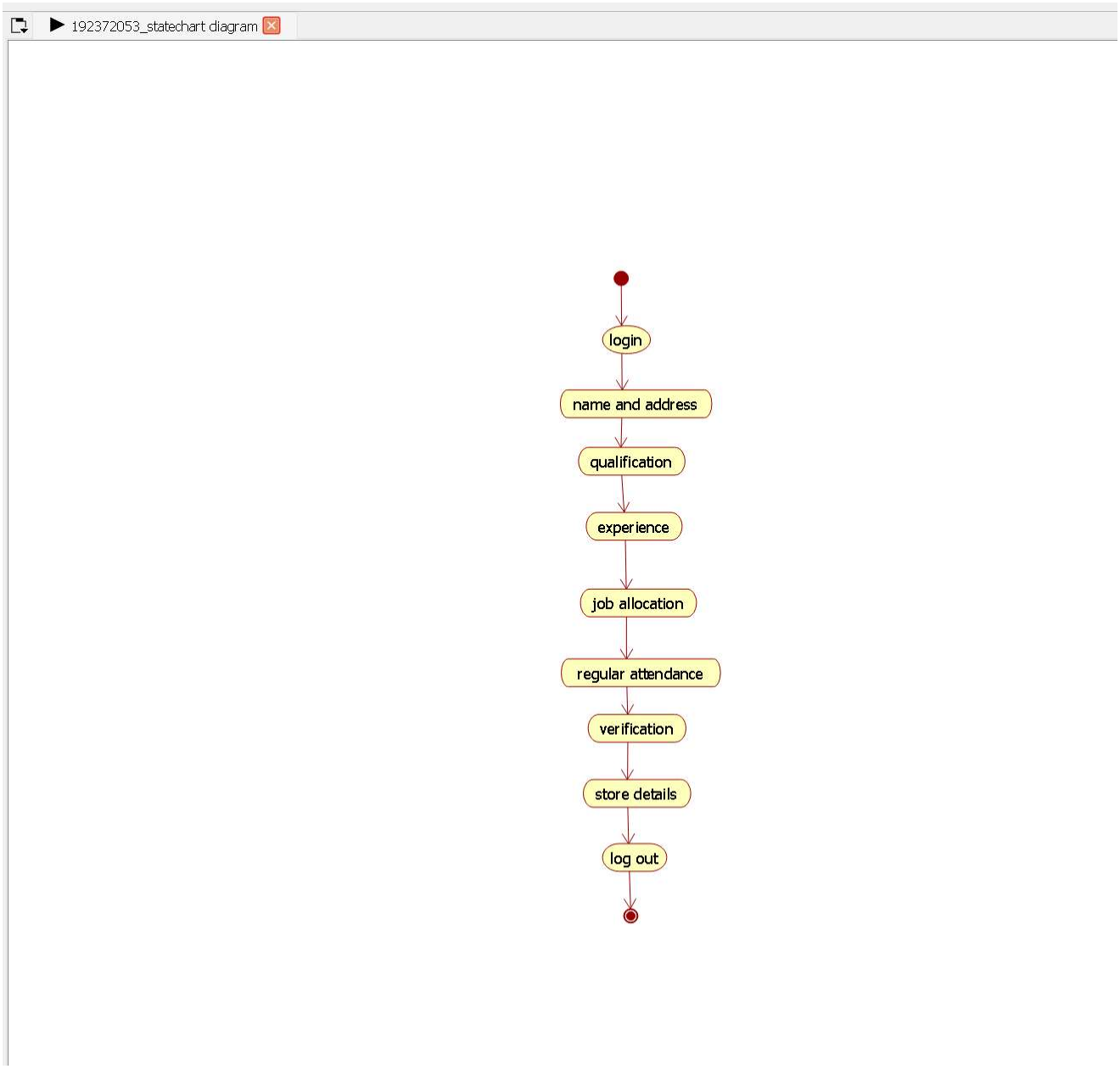
COLLABORATION DIAGRAM:

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



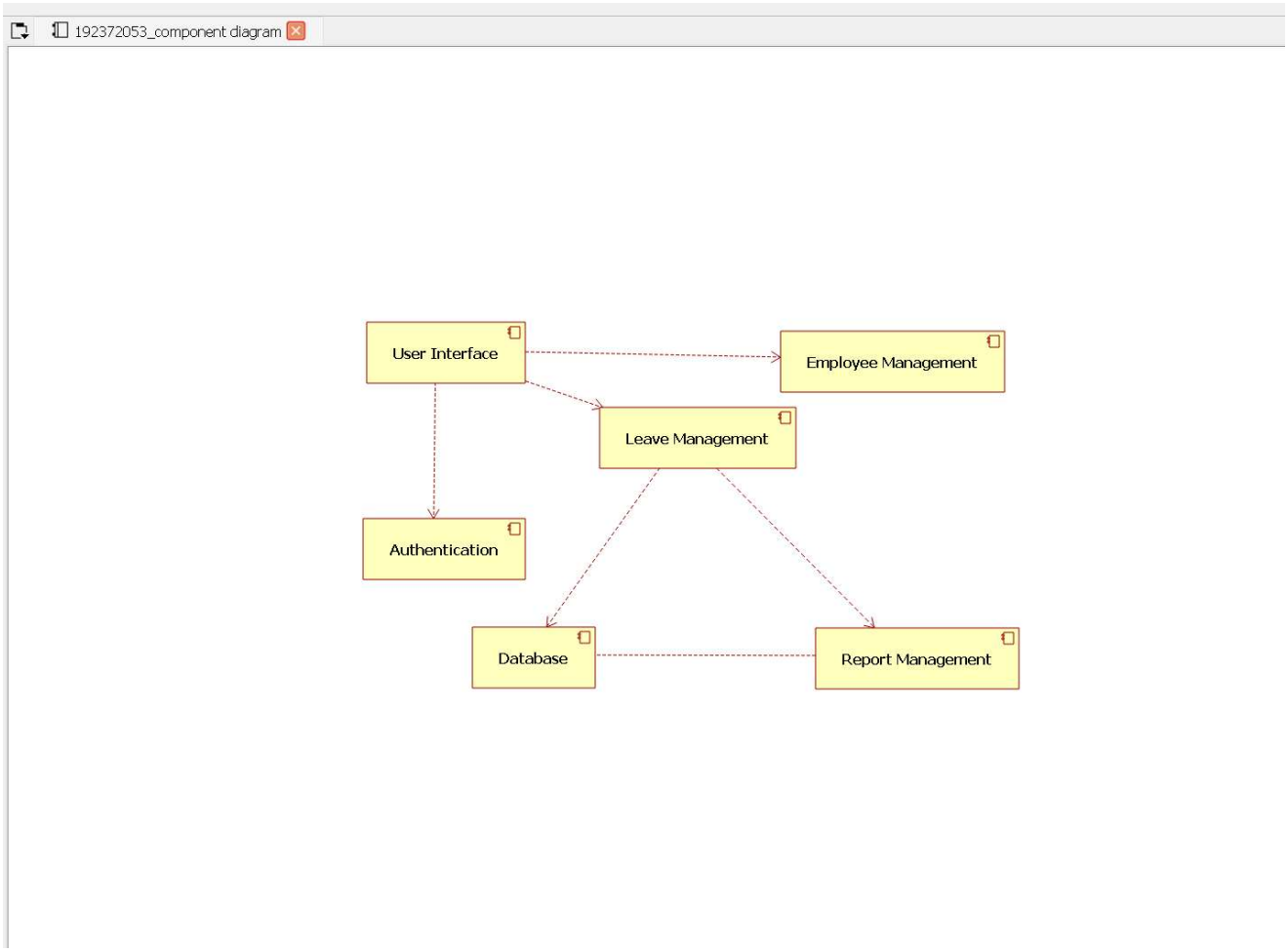
STATECHART DIAGRAM:

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:

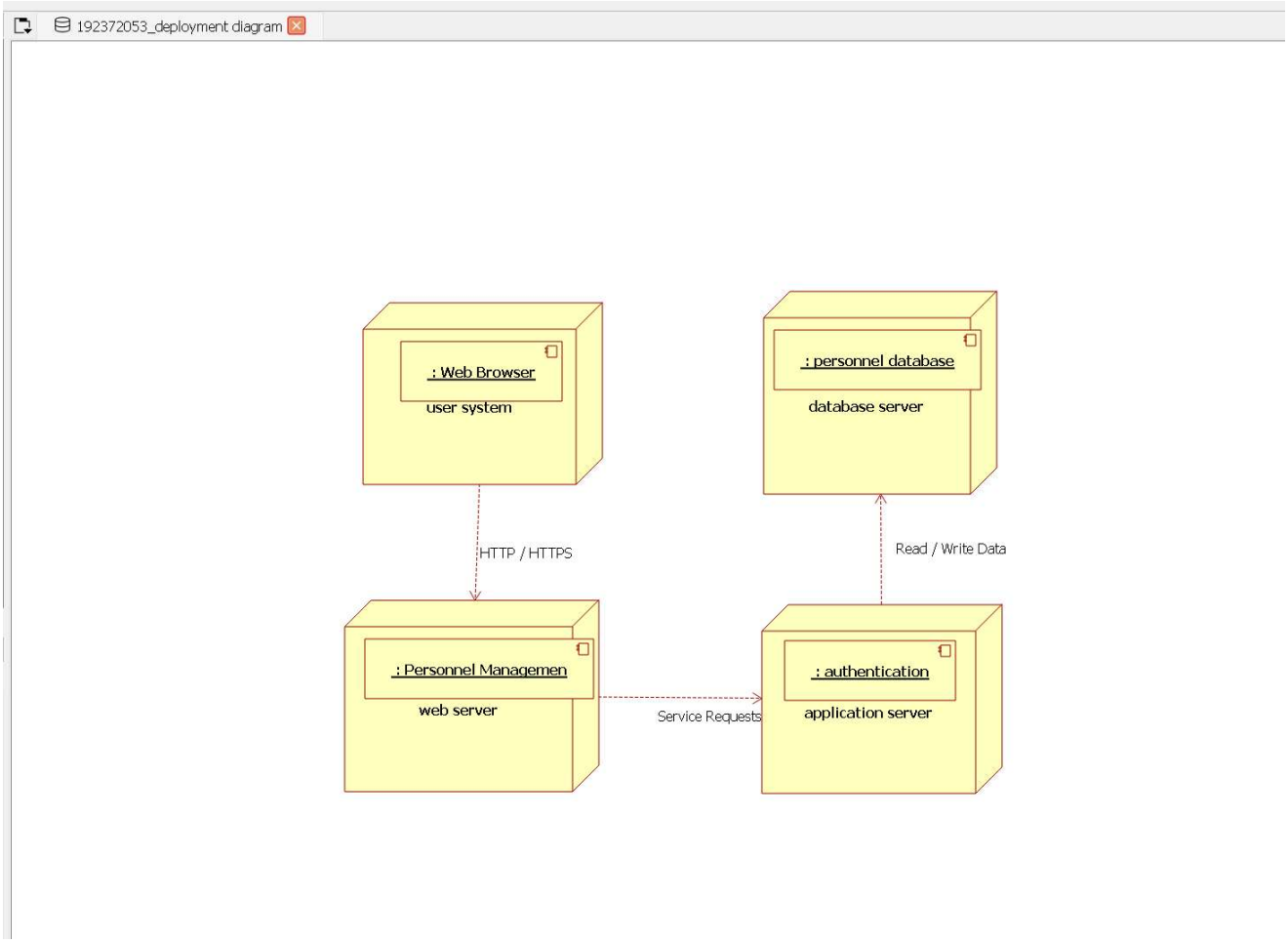
The component diagram is represented by figure dependency and it is a graph of design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a systems. It is represented by boxed figure. Dependencies are represented by communication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



REGISTER NO:

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.

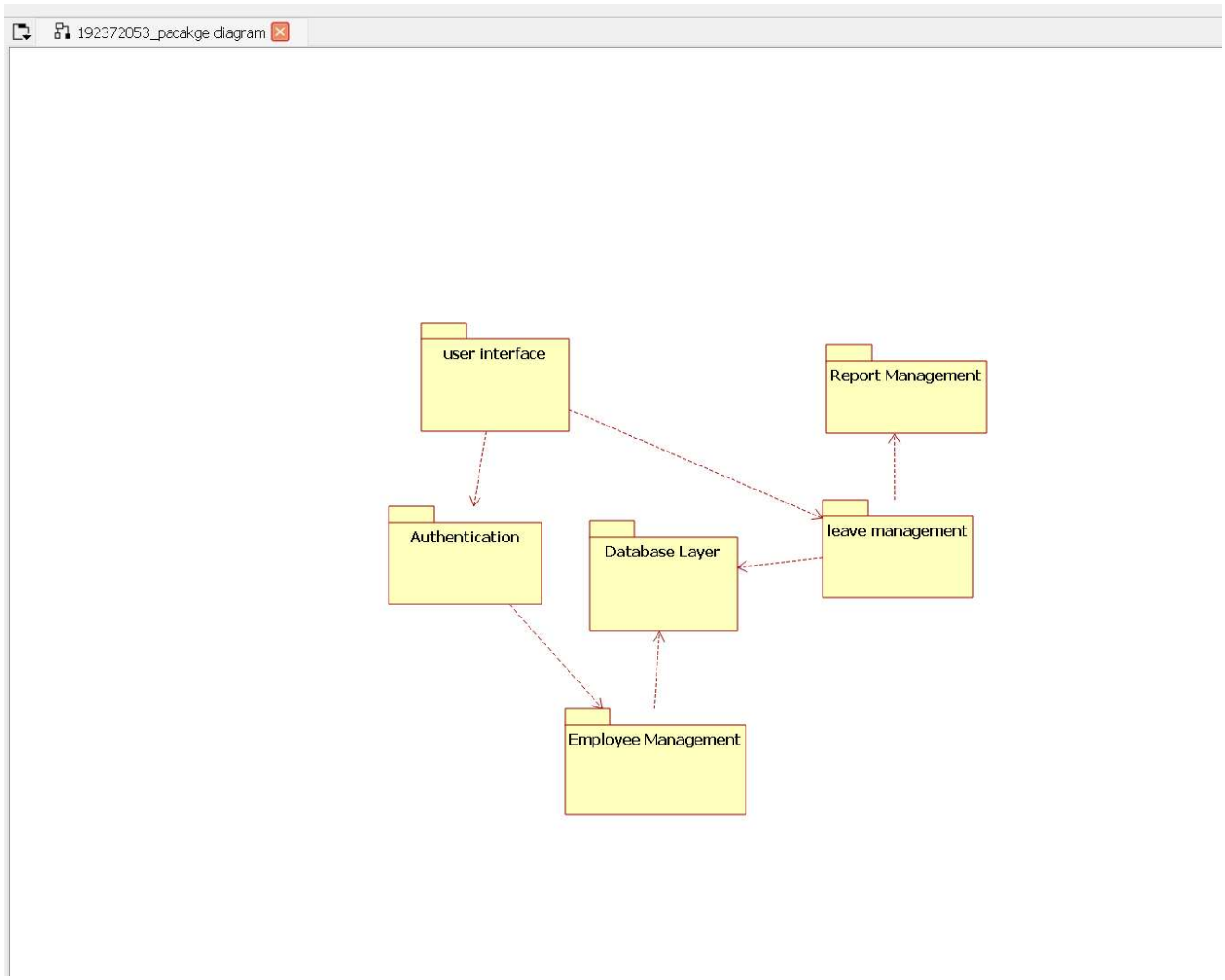


PACKAGE DIAGRAM:

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:

EMPLOYEE:

```
Public class employee
 {
        Public integer employee details;
        Public integer salary;
        Public void leave taken()
         {
         }
        Public void employee()
         {
         }
```

}

HUMAN RESOURCES:

```
Public class HR
{
        Public integer check details;
        Public void loss of pay()
         {
         }
        Public void tax()
         {
         }

        Public void HR()
         {
         }
}
```

CENTRAL MANAGEMENT SYSTEM:

```
Public class central management system {
        Public integer employee name;
        Public integer employee no;
        Public integer details;
        Public void leave taken()
```

```
        {
        }
        Public void tax()
        {
```



```
        }
        Public void loan()
        {
        }
        Public void salary()
        {
        }
}
```

RESULT:

Thus the diagram [usecase, class, activity, sequence, collaboration, state chart, component, deployment, package] for the Software Personnel Management System has been designed, executed and output is verified.

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

AIM:

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

SOFTWARE REQUIREMENTS SPECIFICATION

	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:

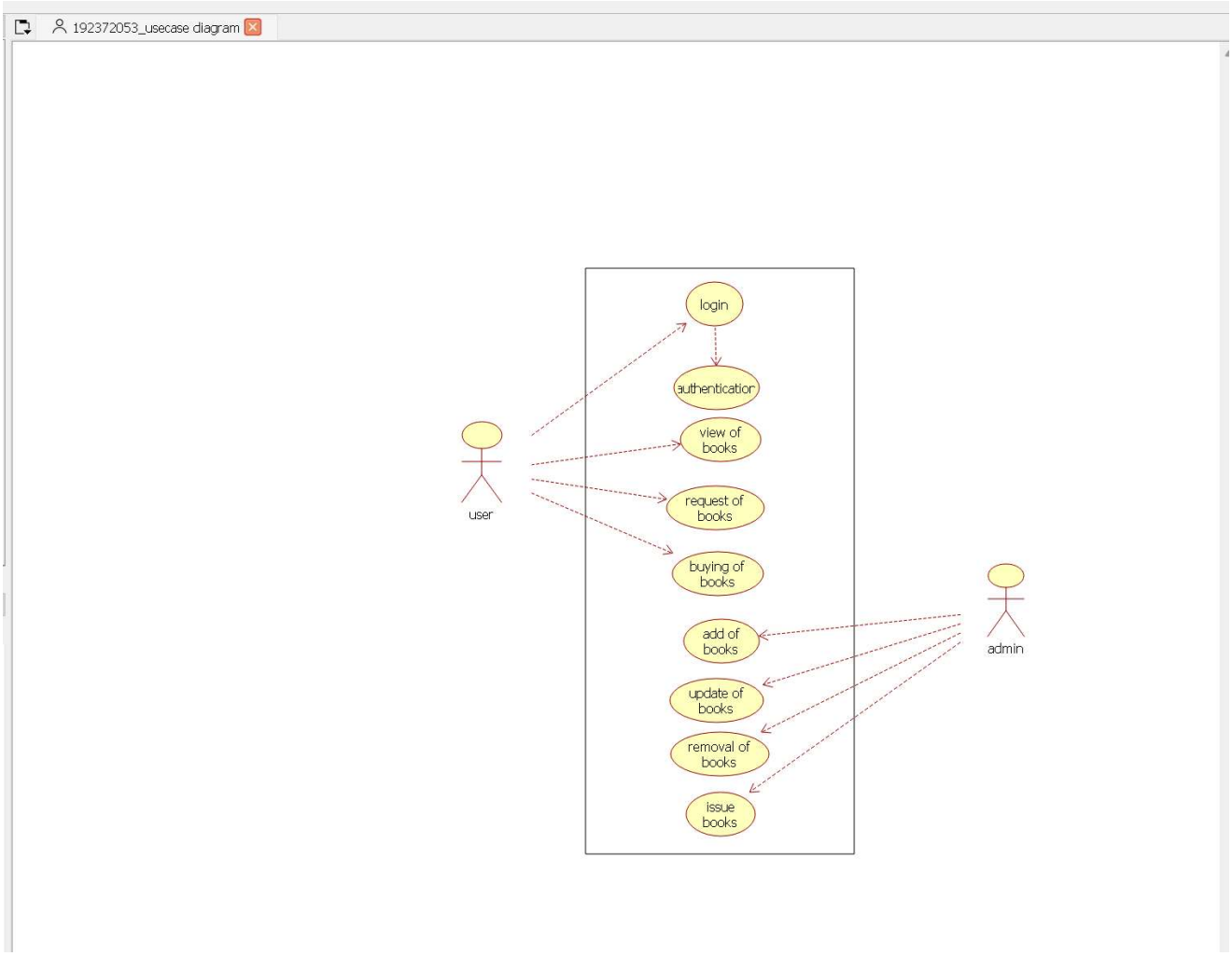
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: user, e-book management

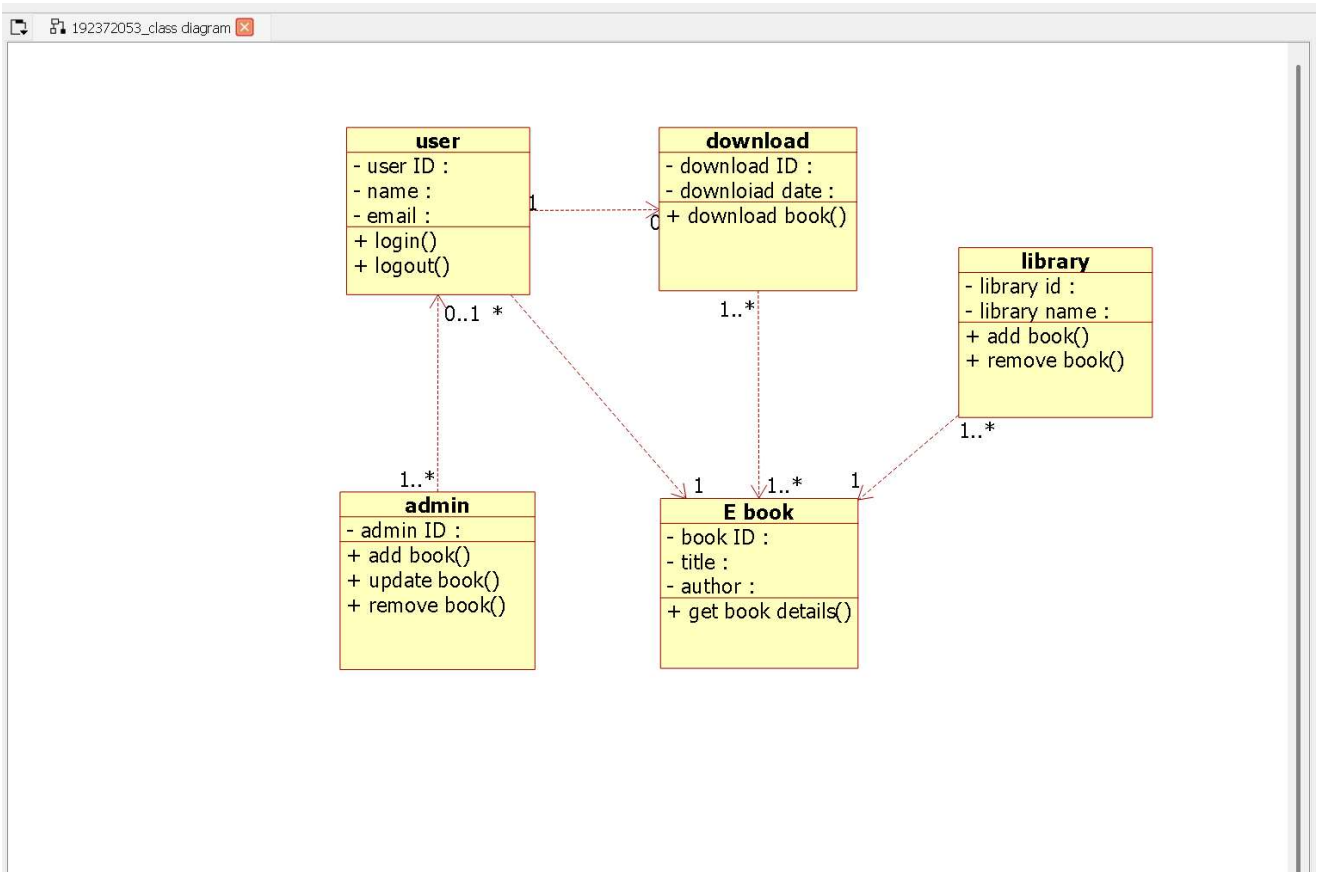
Use case: login ,search books, download ,pay for the books, logout .



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()

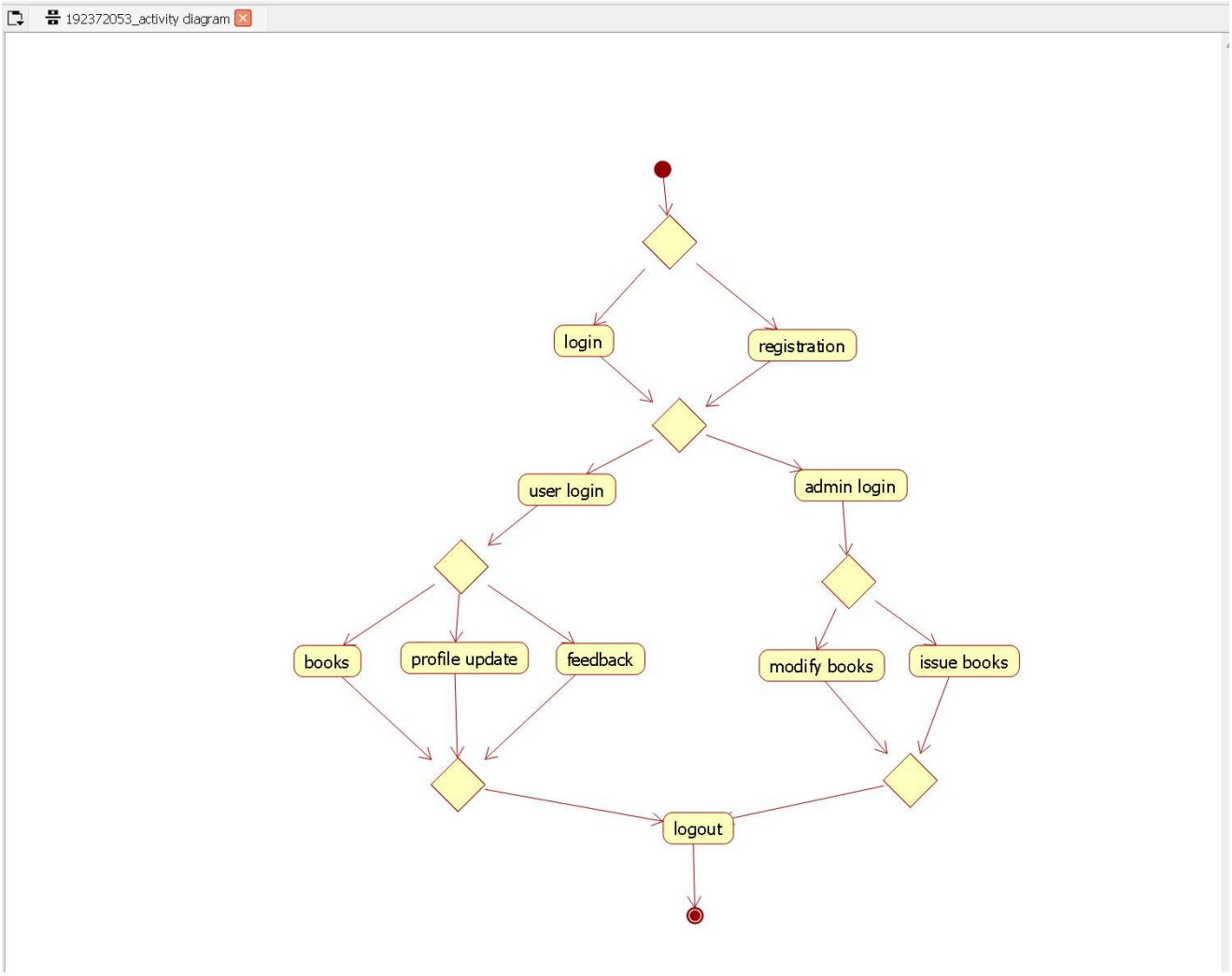


ACTIVITY DIAGRAM:

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

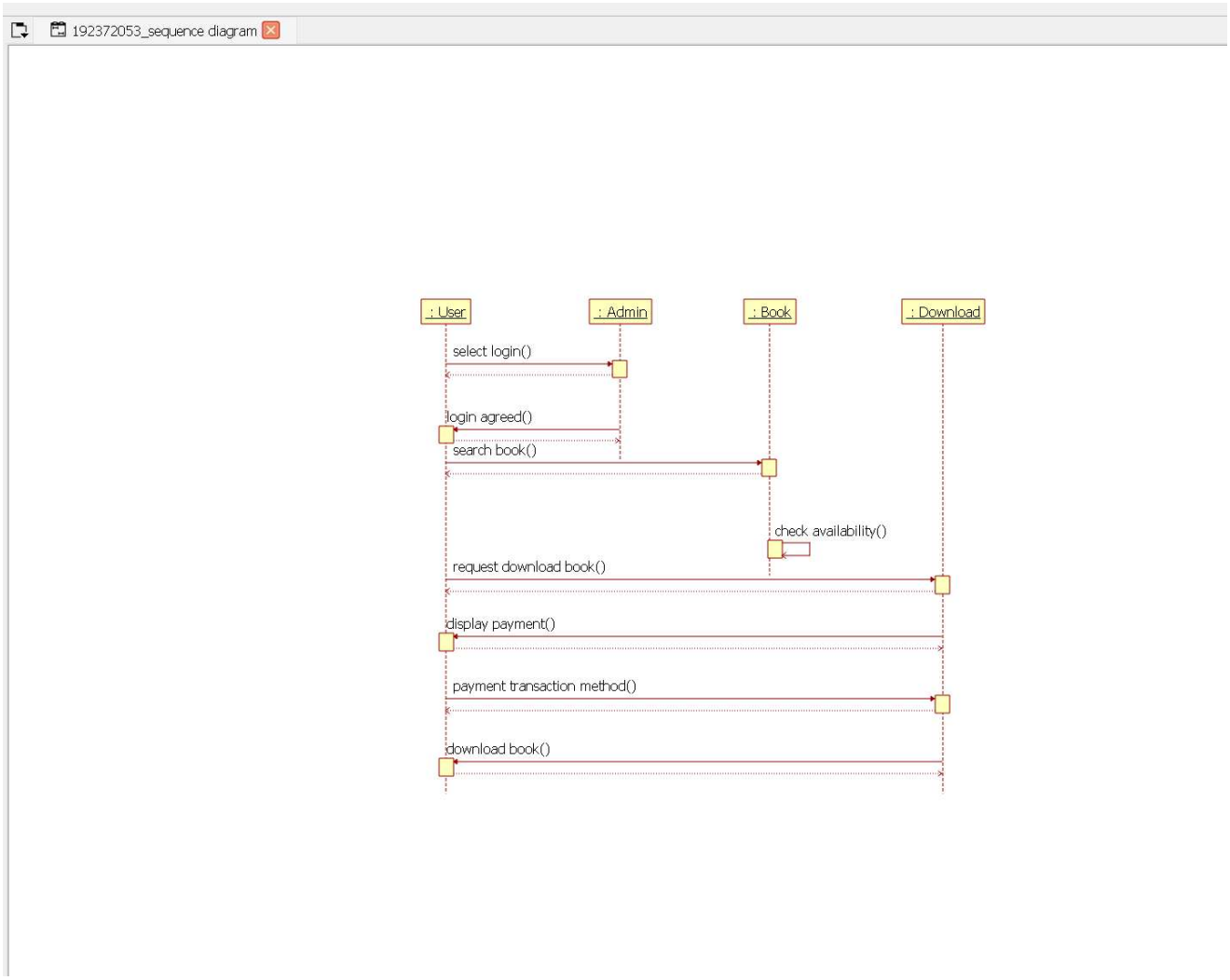
Decision box: check availability



SEQUENCE DIAGRAM:

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

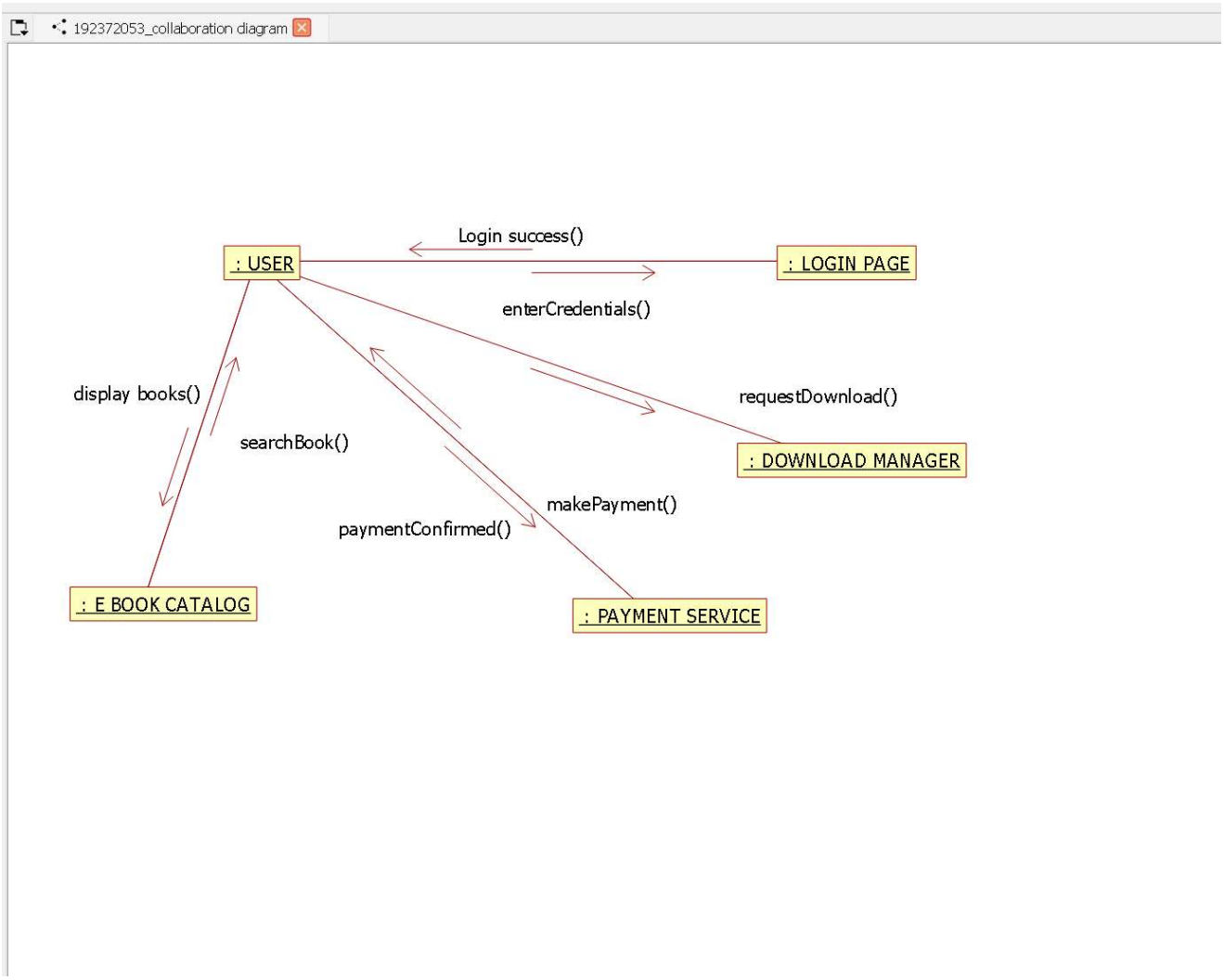
Object: User ,E-book management ,Internet



REGISTER NO:

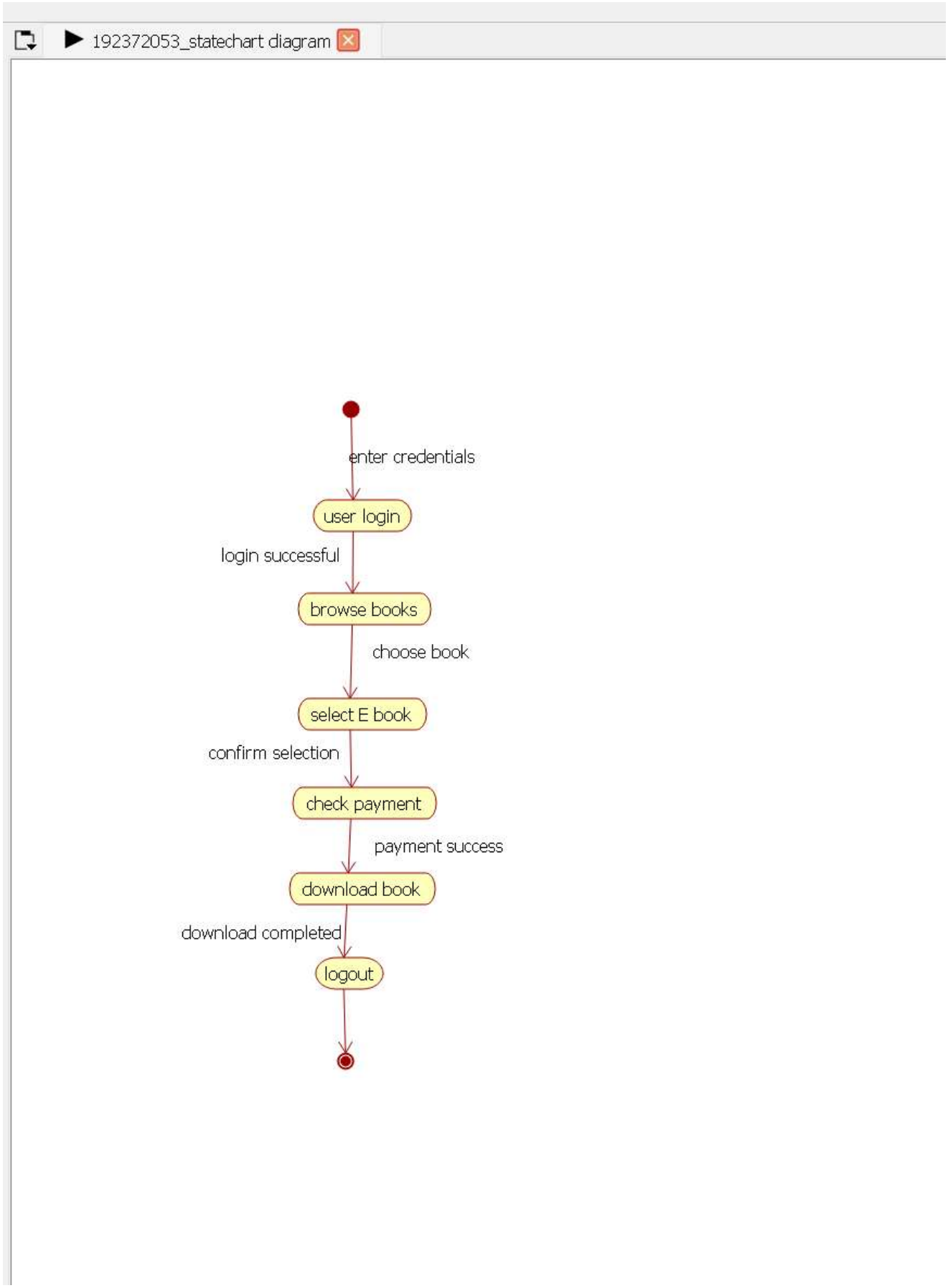
COLLABORATION DIAGRAM:

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



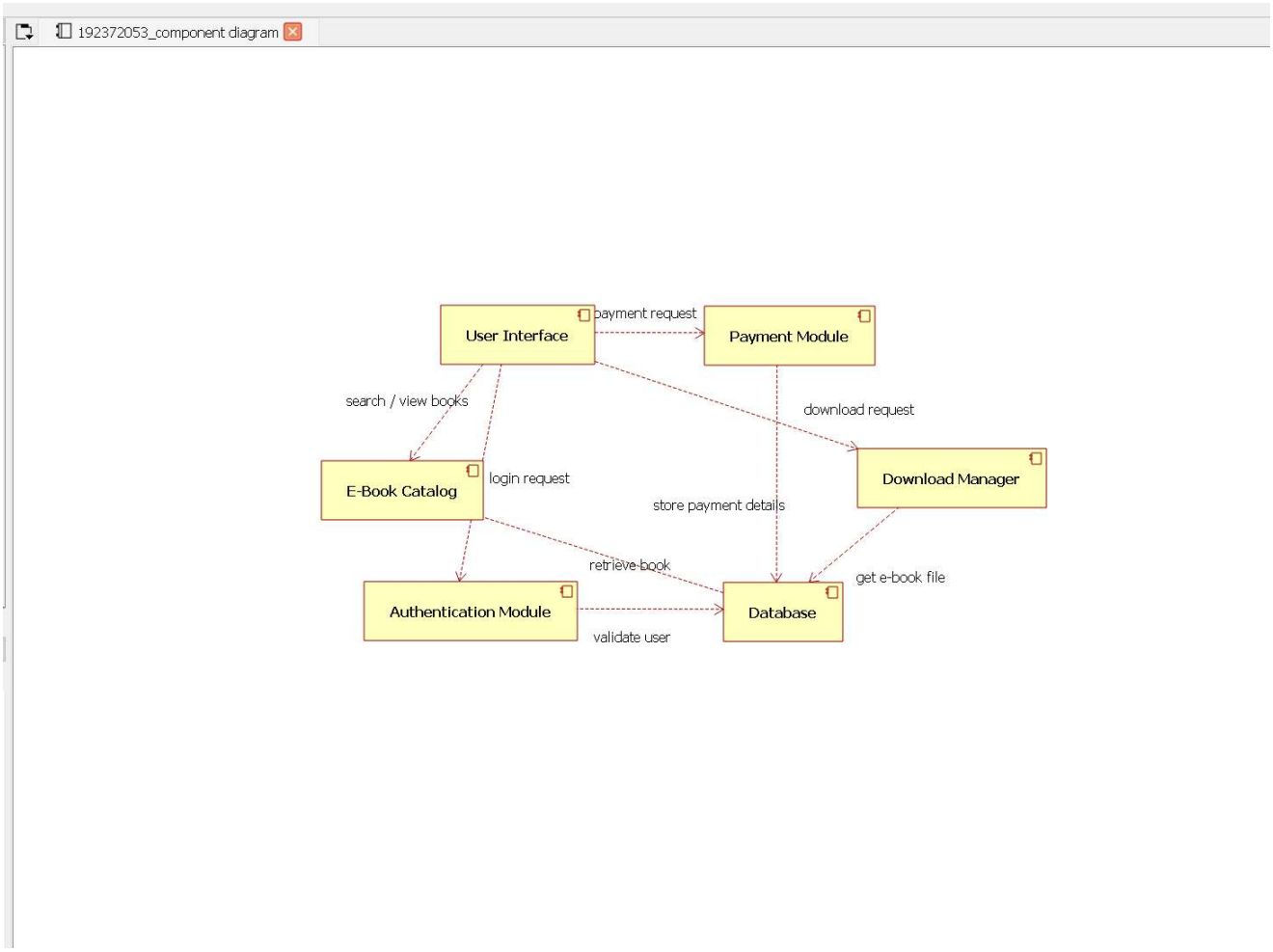
STATECHART DIAGRAM:

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.



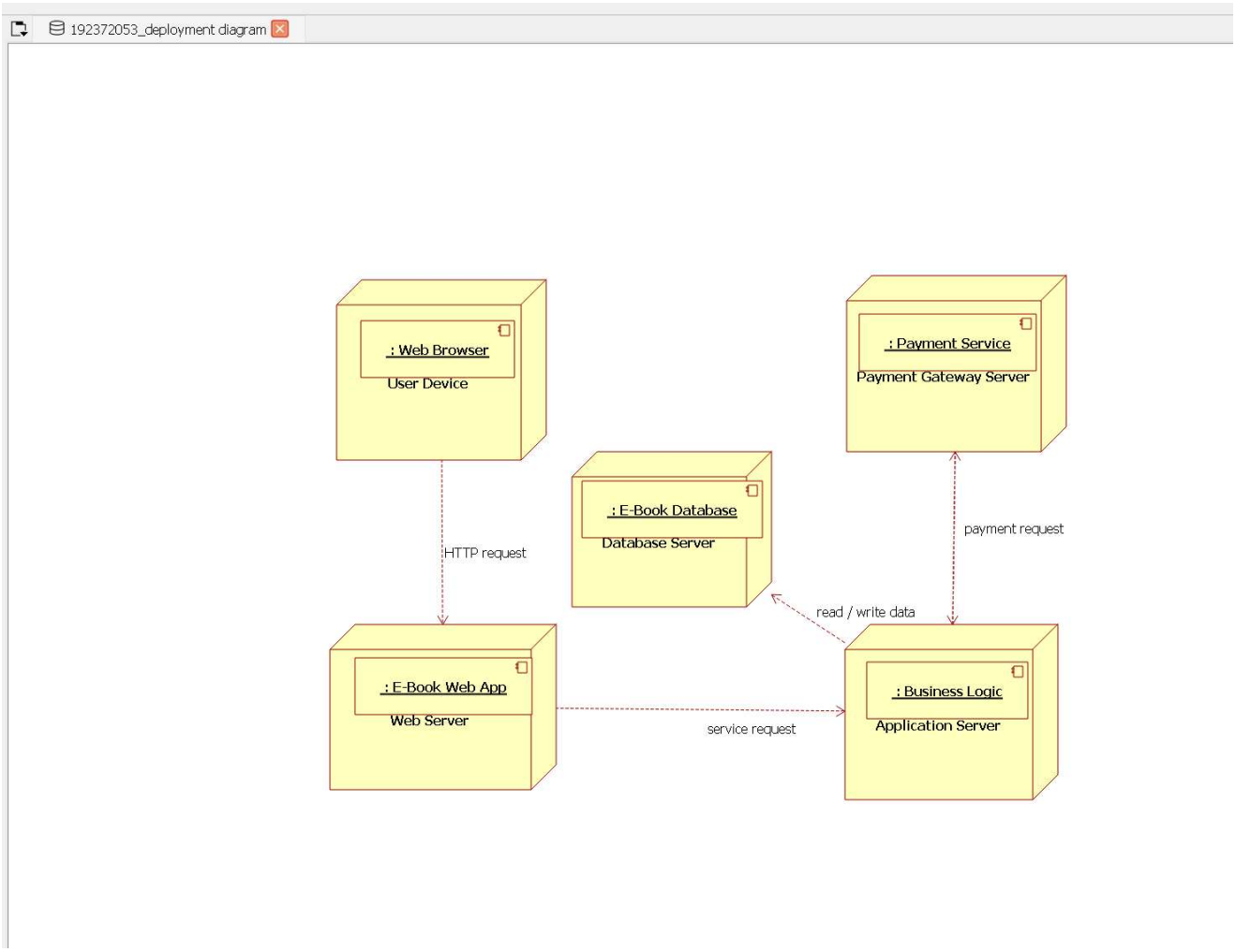
COMPONENT DIAGRAM:

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

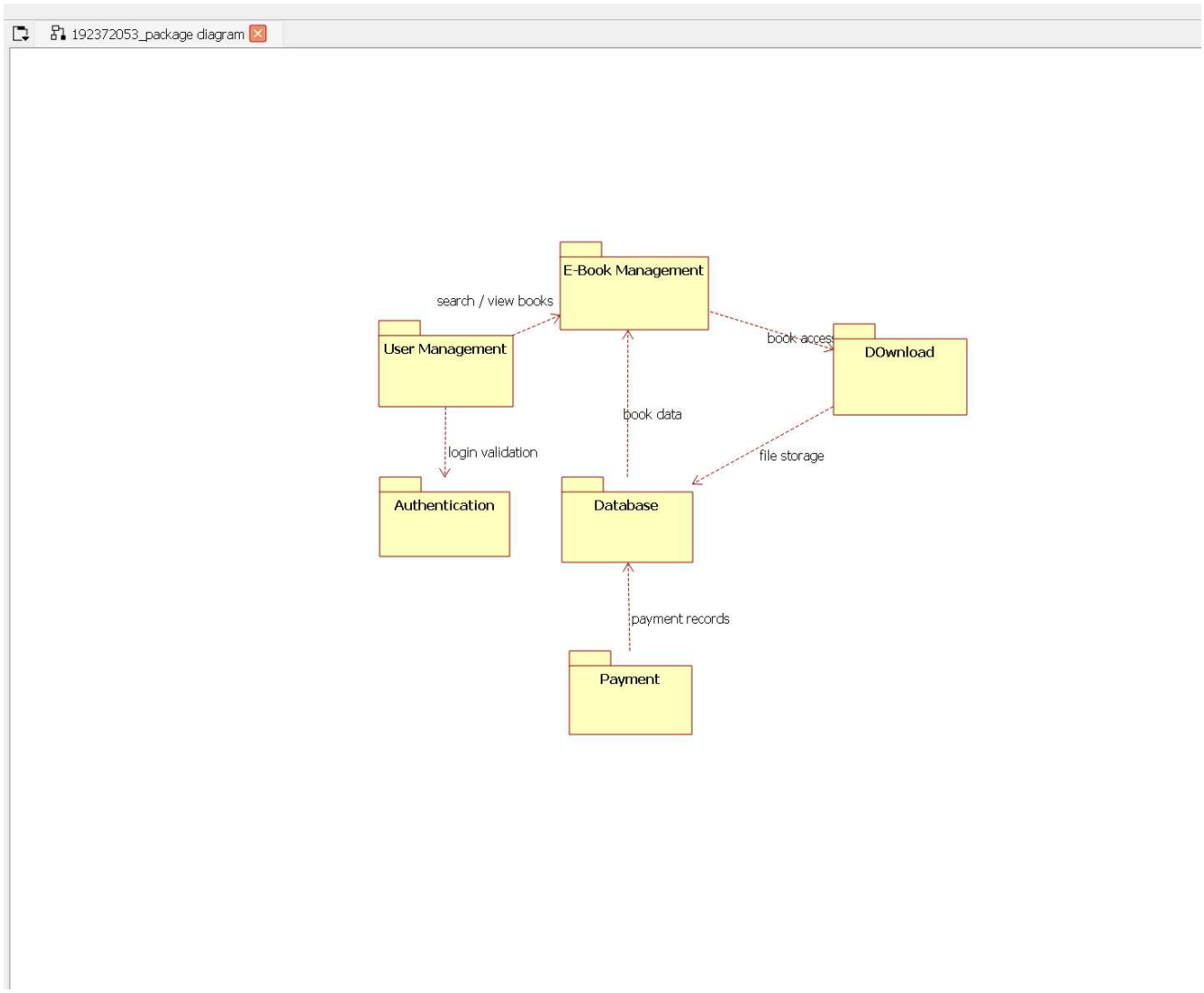


PACKAGE DIAGRAM:

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.

EX:NO:10	RECRUITMENT SYSTEM
DATE:	

AIM:

To draw the diagram[Usecase, Activity, Sequence, Collaboration, Class, StateChart, Component and Deployment, package] for recruitment system.

SOFTWARE REQUIREMENTS SPECIFICATION:

	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

1.3 PROJECT DESCRIPTION:

This system is designed to recruit the particular job to the person in a company .It was controlled by the central management system to manage the details of the particular candidate that one has to be recruited for a company.

1.4 REFERENCES:

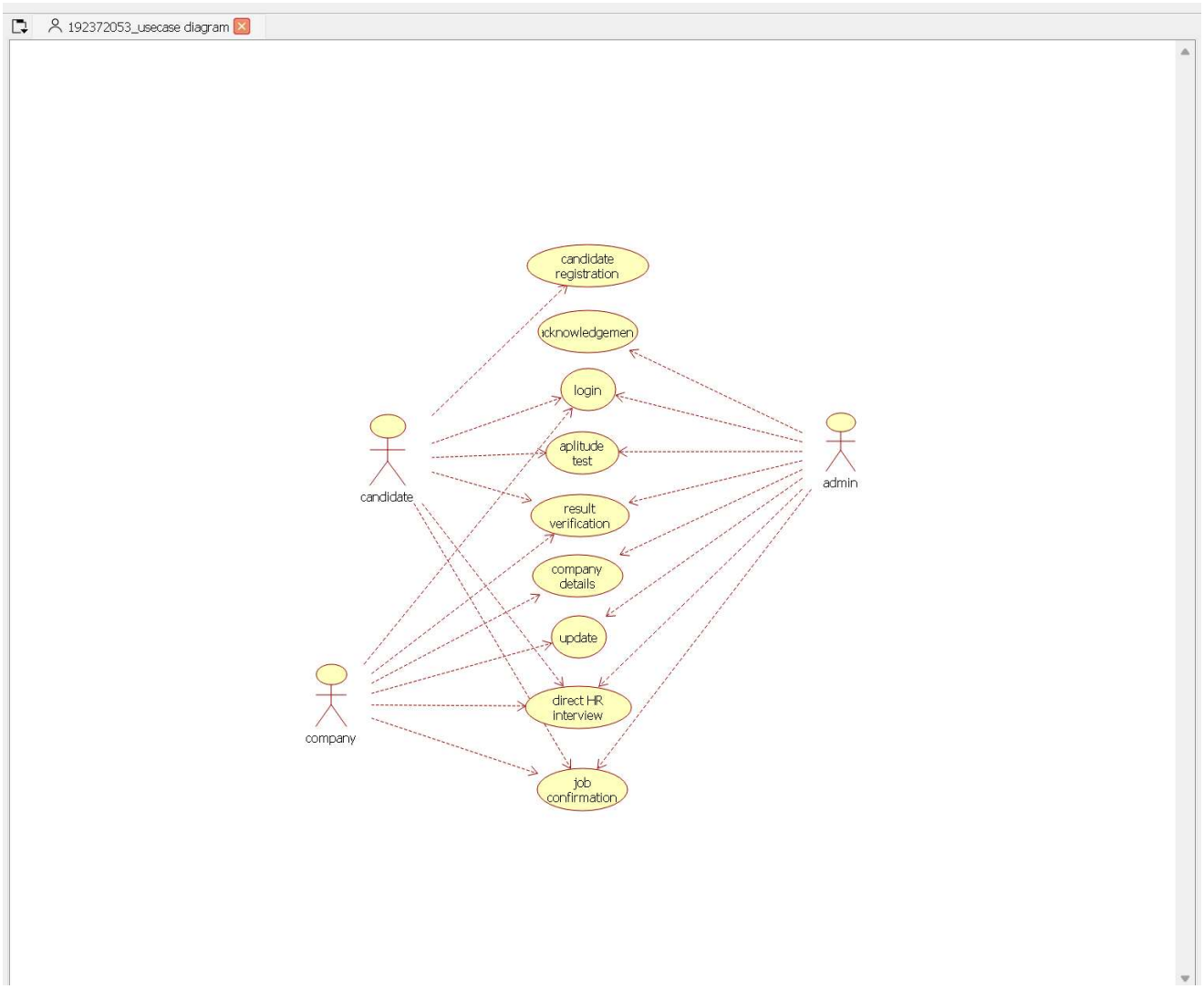
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Applicant, HR, Central management system.

Use case: Aptitude, Group discussion, Technical skills, Personal specification, Short list, Result

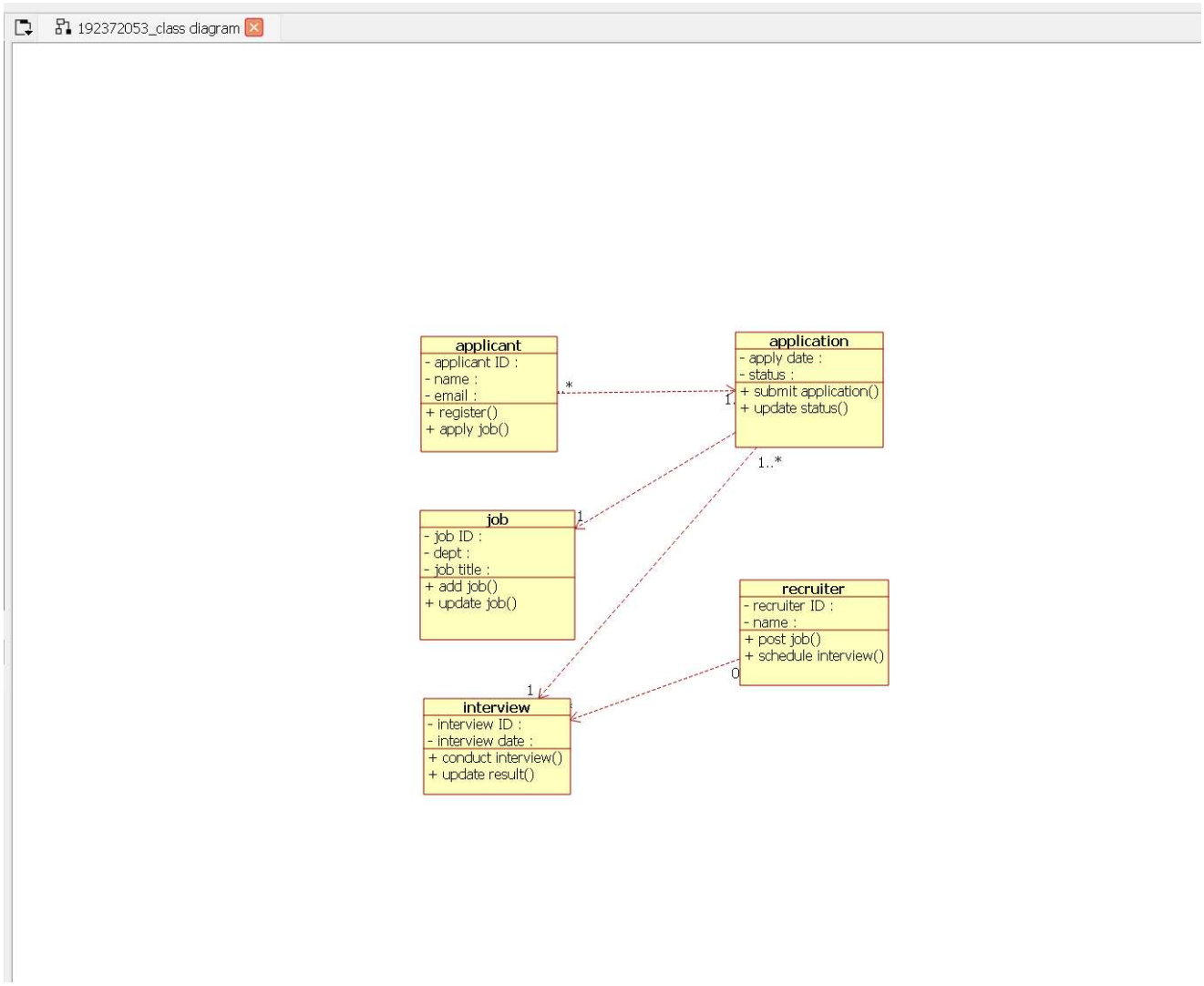


REGISTER NO:

CLASS DIAGRAM:

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

CLASSES	ATTRIBUTES	OPERATIONS
Candidate	Name, qualification	Verify()
HR	Verification, resume	Select()
Central system	Store, update	Update()

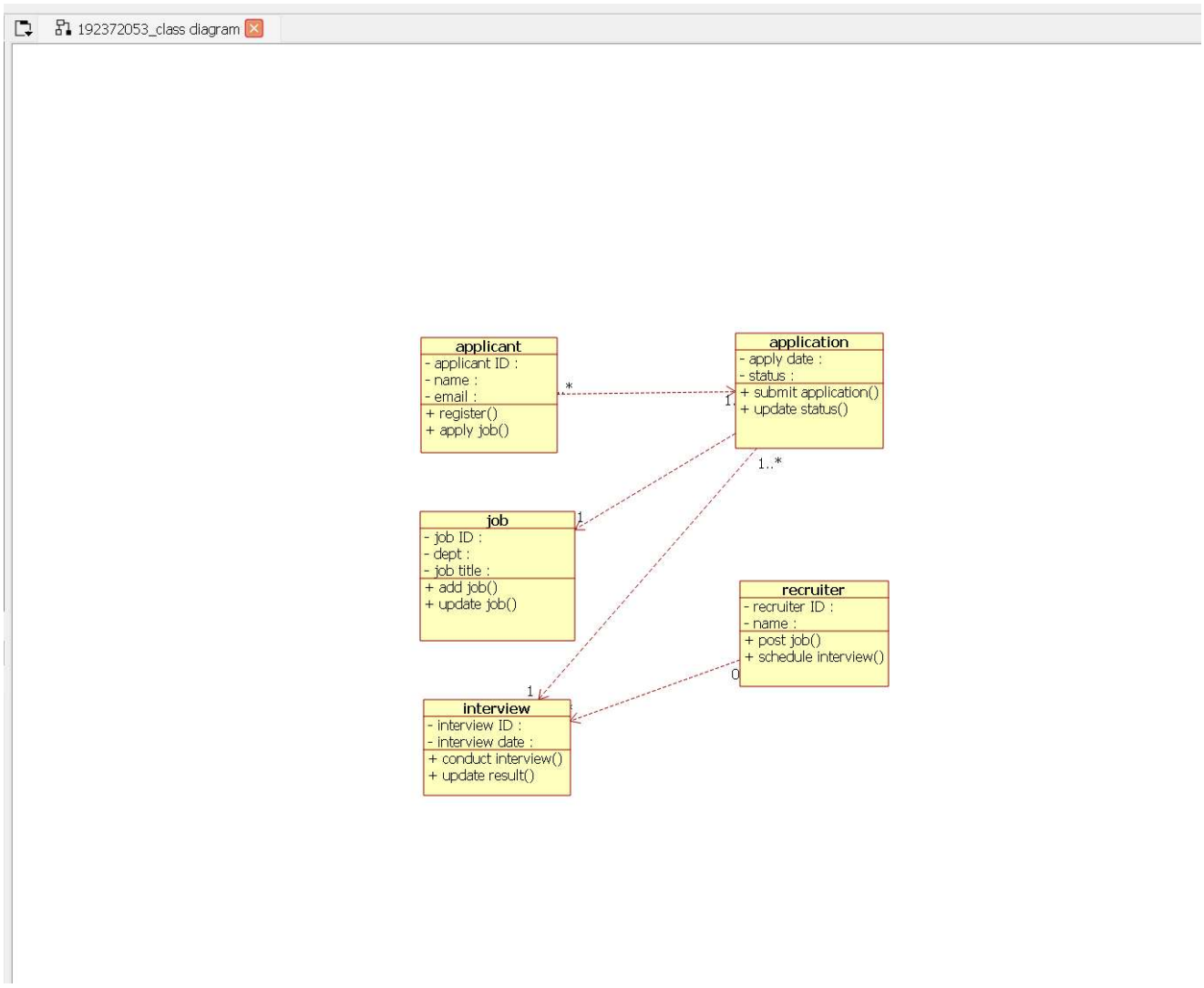


ACTIVITY DIAGRAM:

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

Activities: Aptitude, Group discussion ,Technical skills,HR

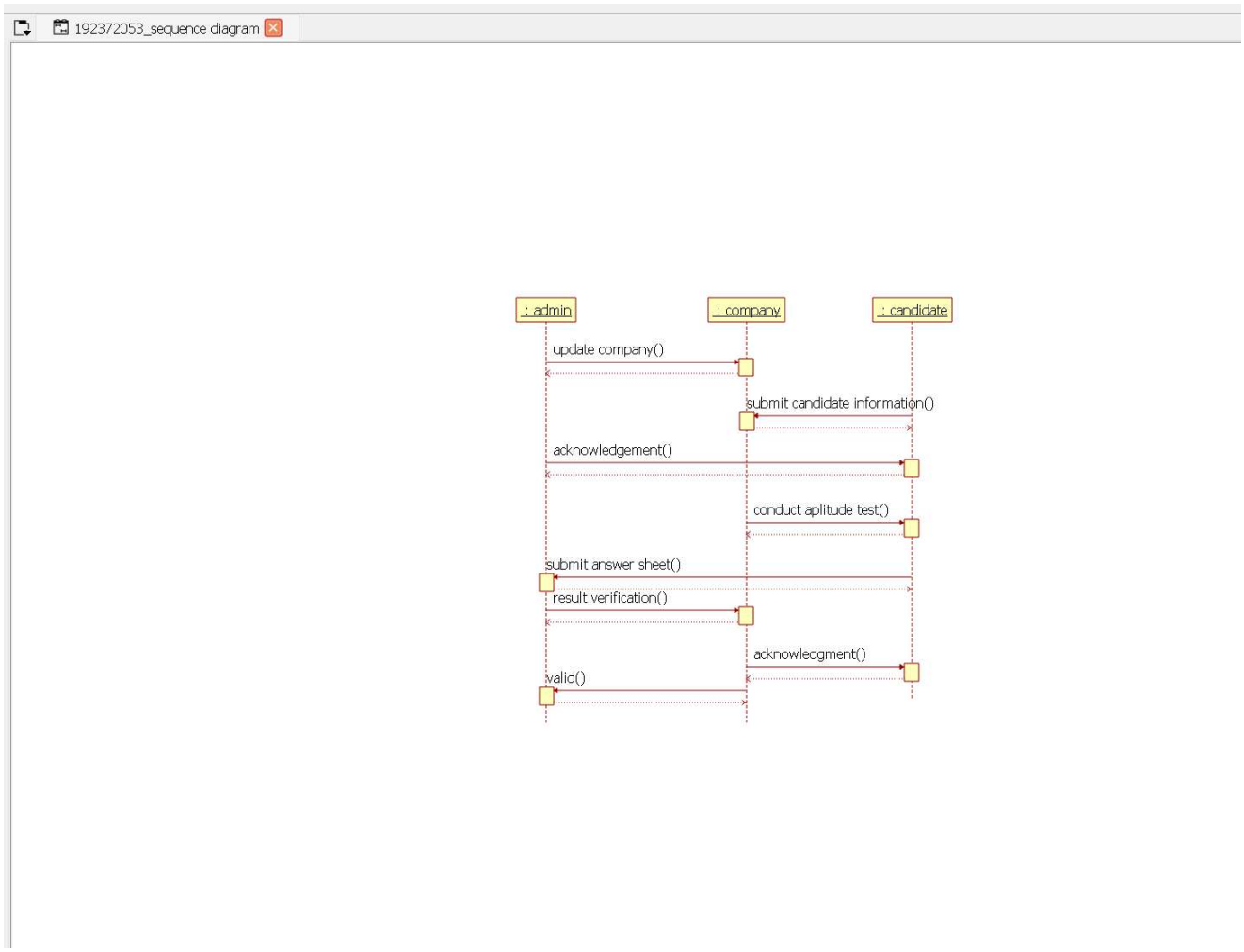
Decision box: Verification of the qualities



SEQUENCE DIAGRAM:

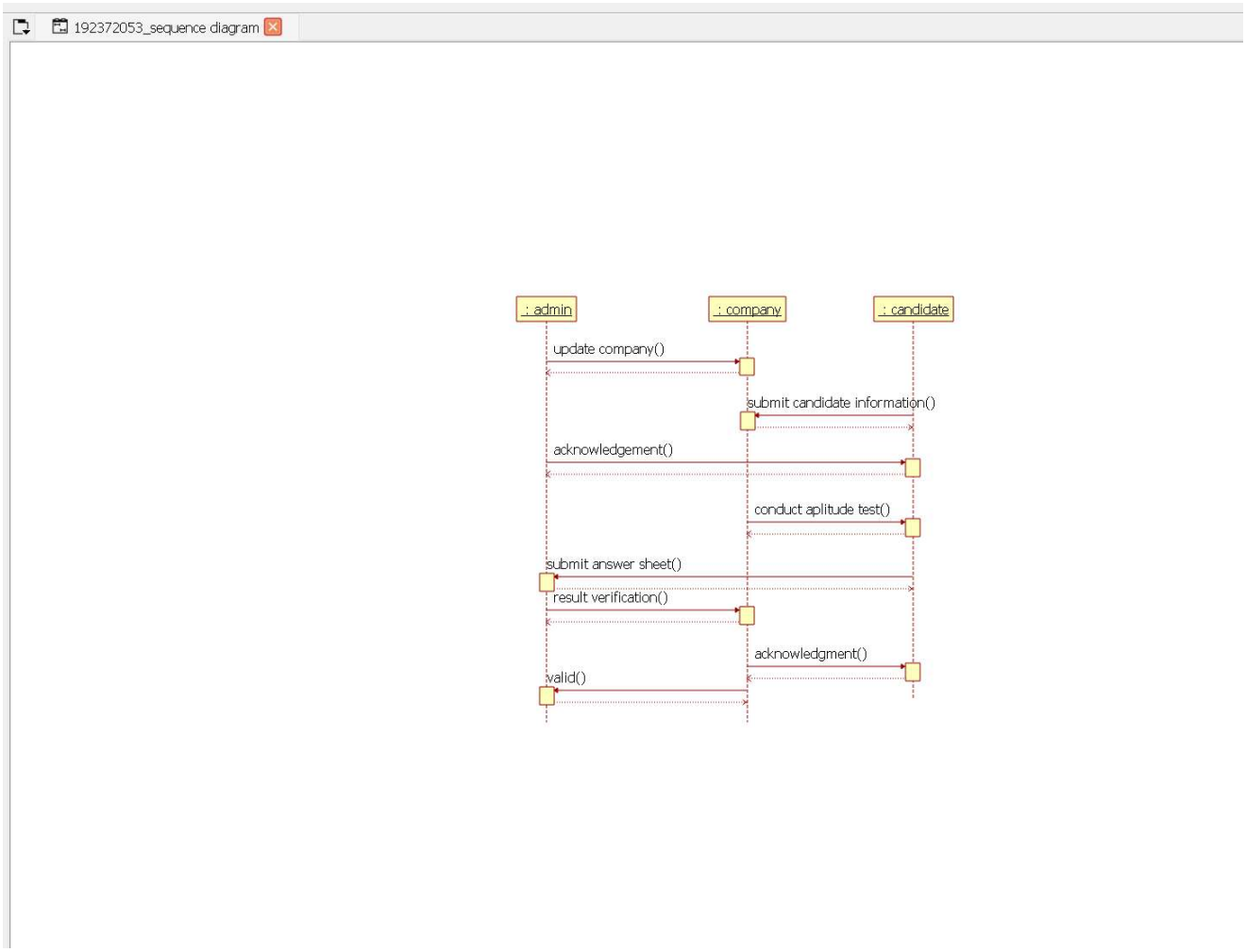
This diagram consists of the objects, messages and return messages.

Object: Candidate, HR, Central system



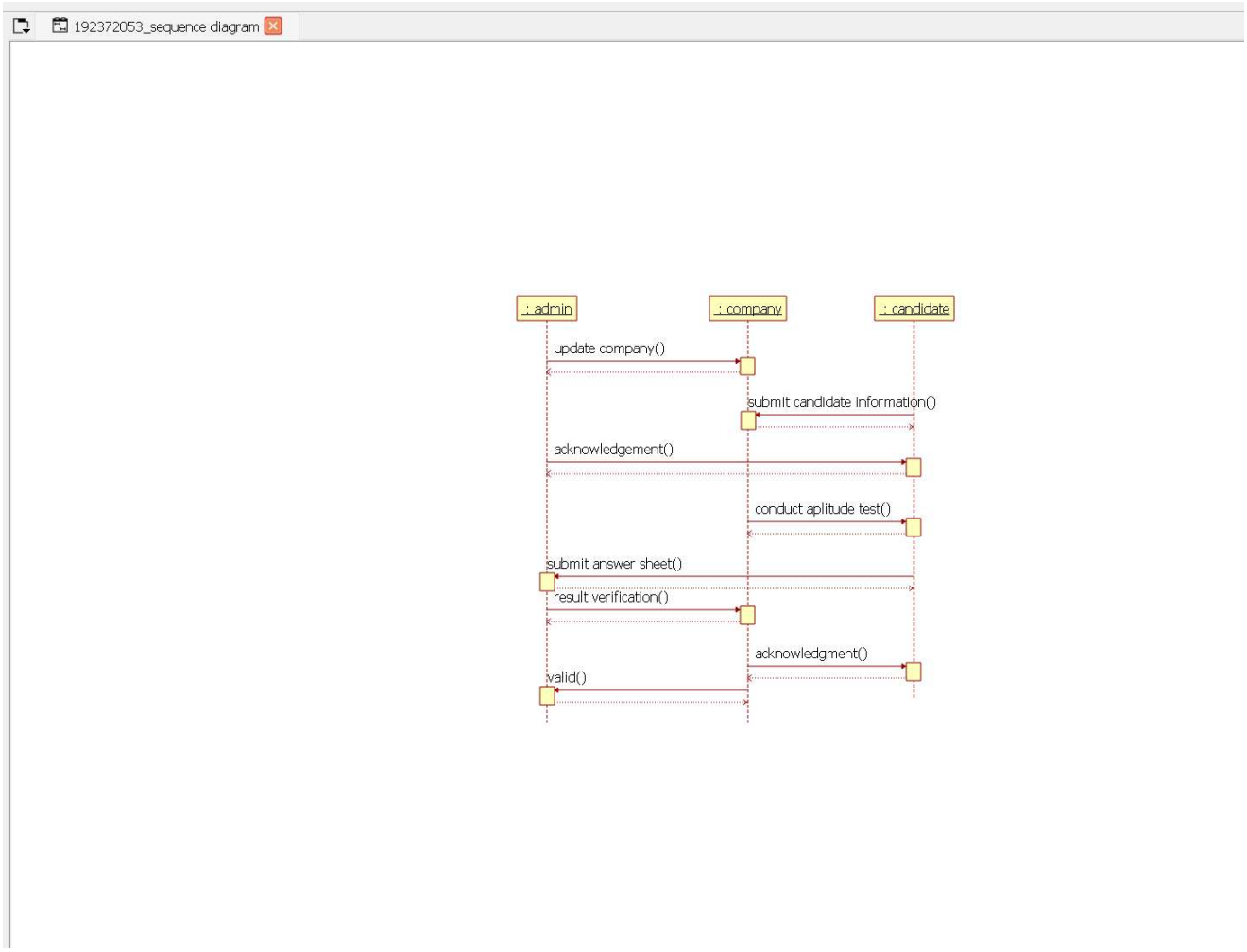
COLLABORATION DIAGRAM:

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



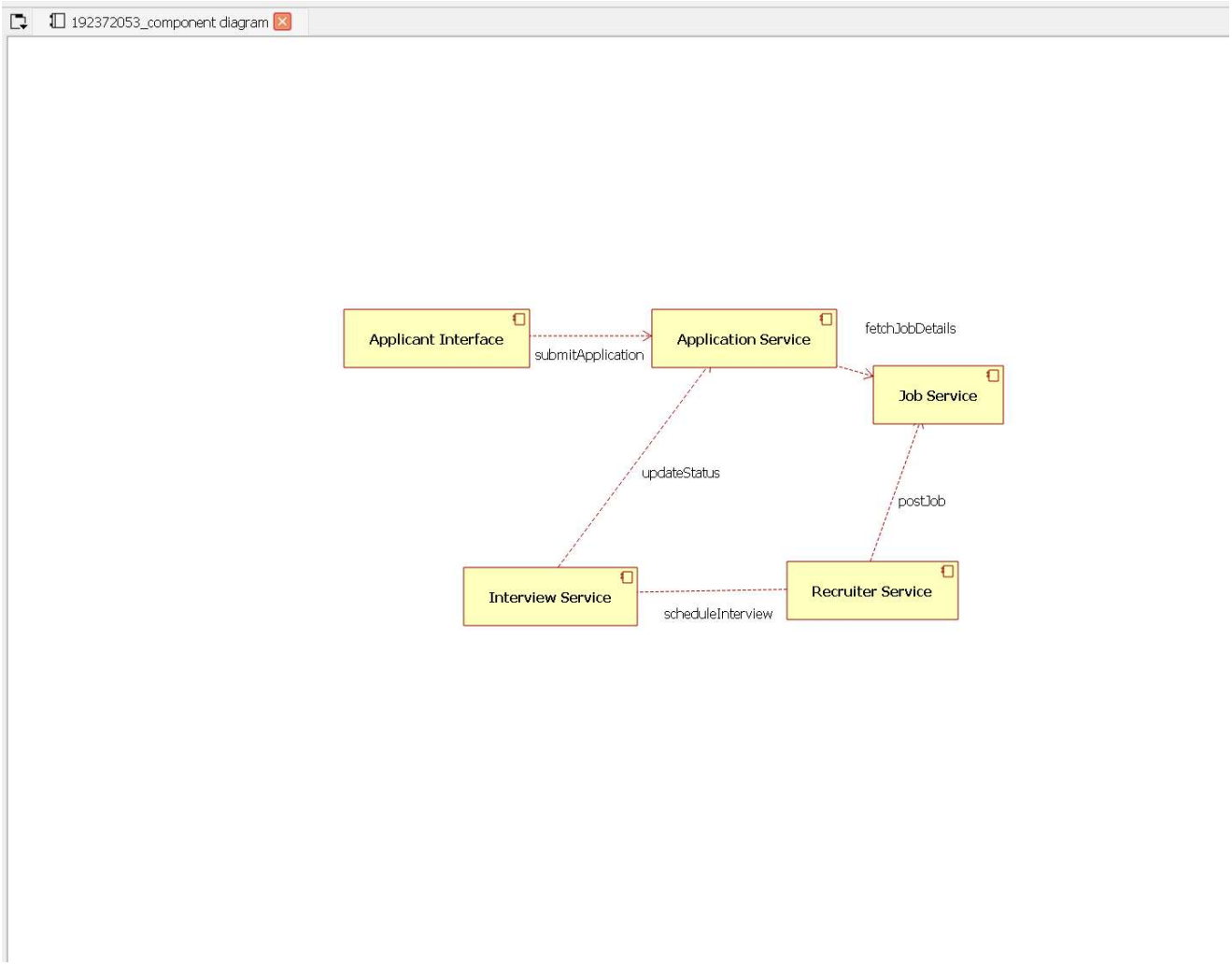
STATE CHART DIAGRAM:

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



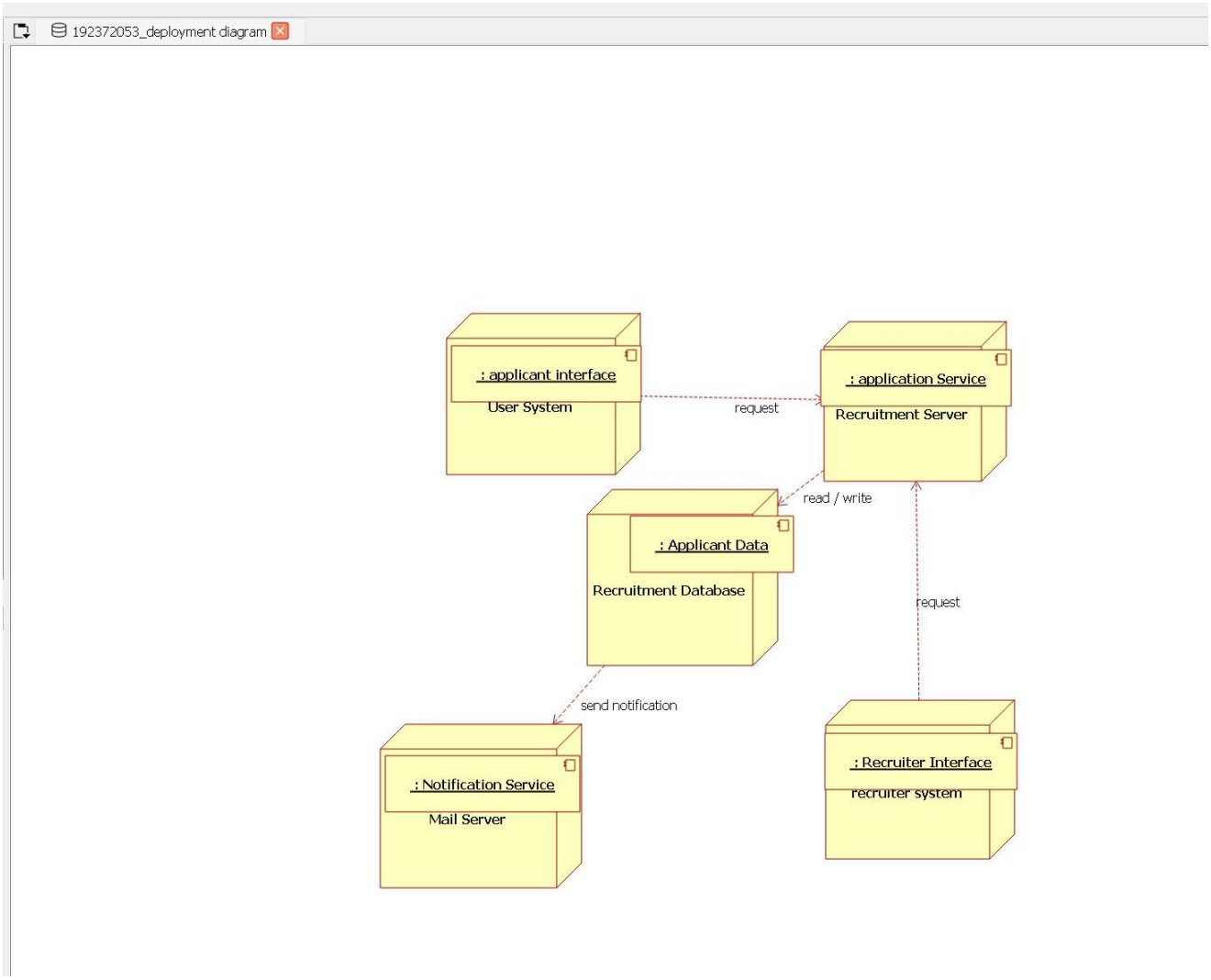
COMPONENT DIAGRAM:

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



REGISTER NO:

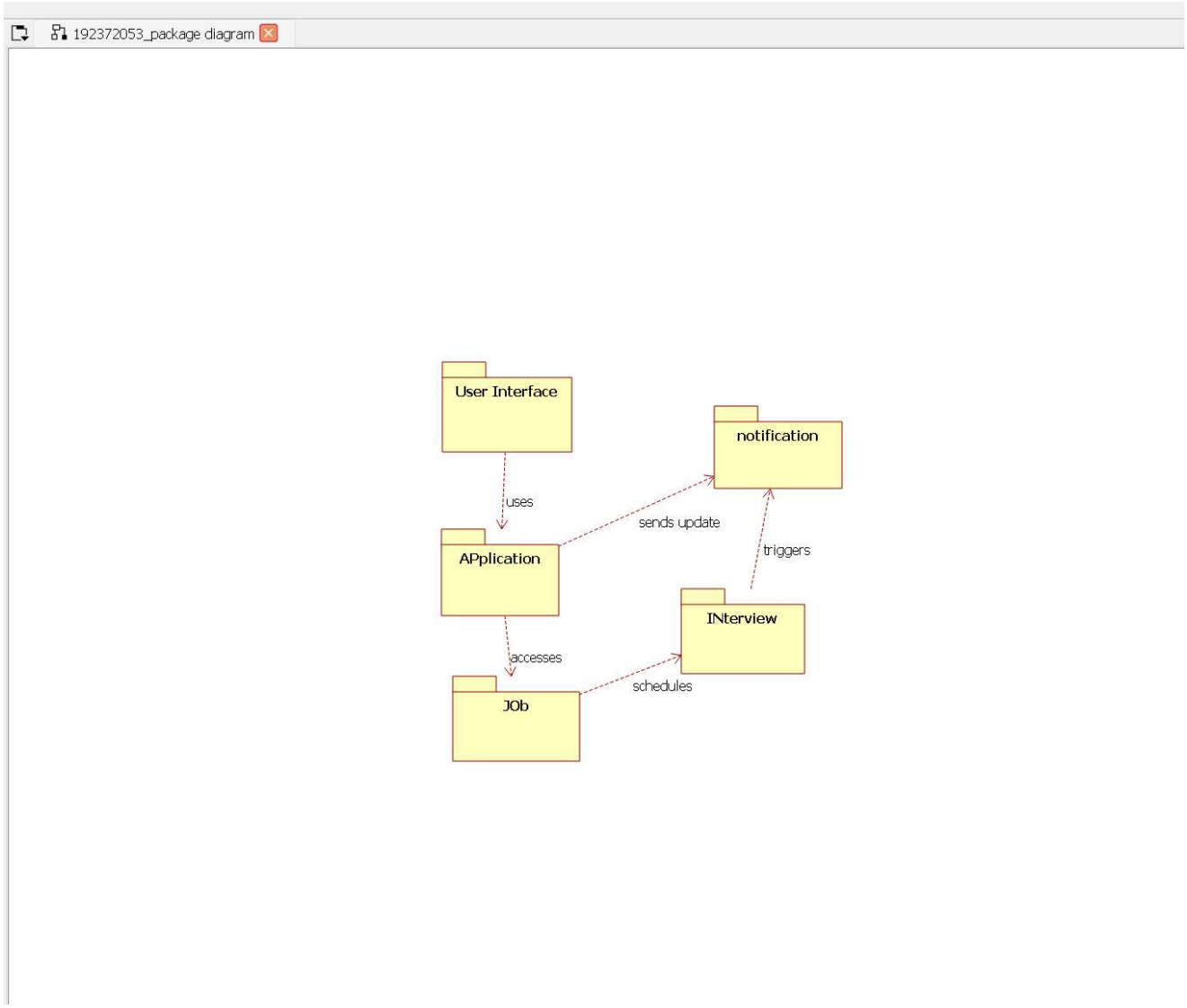
artifacts to nodes according to the Deployments defined between them. It is represented by 3-dimentional box. Dependencies are represented by communication association

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:

CENTRAL SYSTEM:

```
Public class central system
 {
        Public integer store;
        Public integer update;
        Public void central management system()
         {
         }
}
```

HR:

```
Public class HR
 {
        Public integer verification;
        Public integer resume;
        Public void HR()
         {
         }
```



```
}
```

CANDIDATE:

```
Public class candidate
 {
        Public integer name;
        Public integer operation;
        Public integer qualification;
        Public void verify()
         {
         }
        Public void candidate()
         {
         }
}
```

RESULT:

To draw the diagram [Use case, Activity, Sequence, Collaboration, Class, State Chart, Component and Deployment, package] for recruitment system has been designed and output is verified.

EX:NO:11	**CONFERENCE MANAGEMENT SYSTEM**
DATE:	

AIM:

To draw the diagrams [use case, activity, sequence, collaboration, class, component, deployment, package] for Conference management system

SOFTWARE REQUIREMENTS SPECIFICATION

	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

The Conference Management System is an online website in which candidate can submit the paper and register themselves and then attend the conference. The paper will be reviewed. The details of the conference, date and time will be made available to them through the website. After getting the confirmation details the candidate should submit the revised and camera ready paper. Then the registration process will be done.

1.3 PROJECT DESCRIPTION:

This software is designed to manage the details of the process that will be taken place in the conference in a place. It works along with the organizer, who arranges all these program and central management system, which consists of the all the details of the member who participates in the presentation

1.4 REFERENCES:

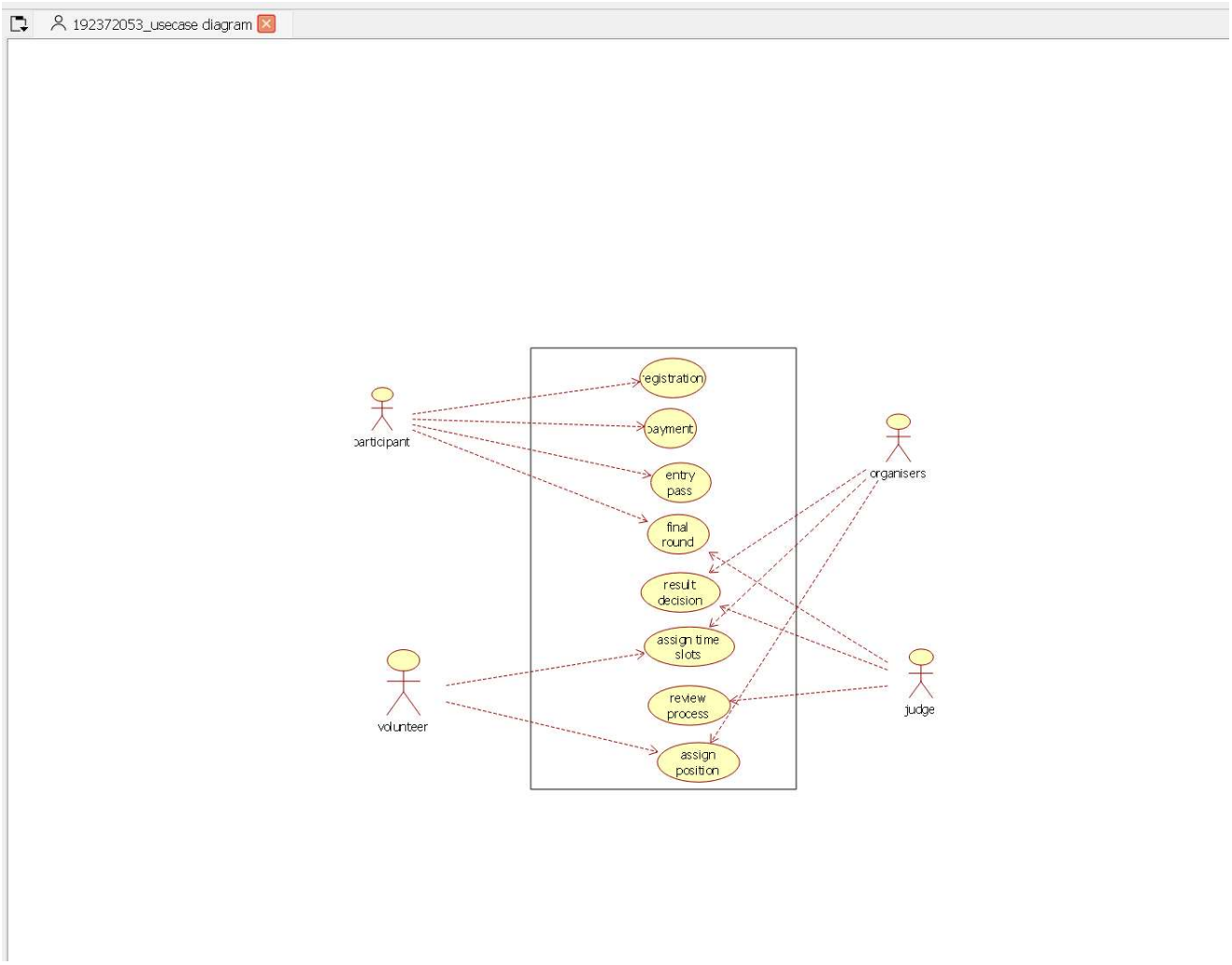
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Member, Organizer, Central system

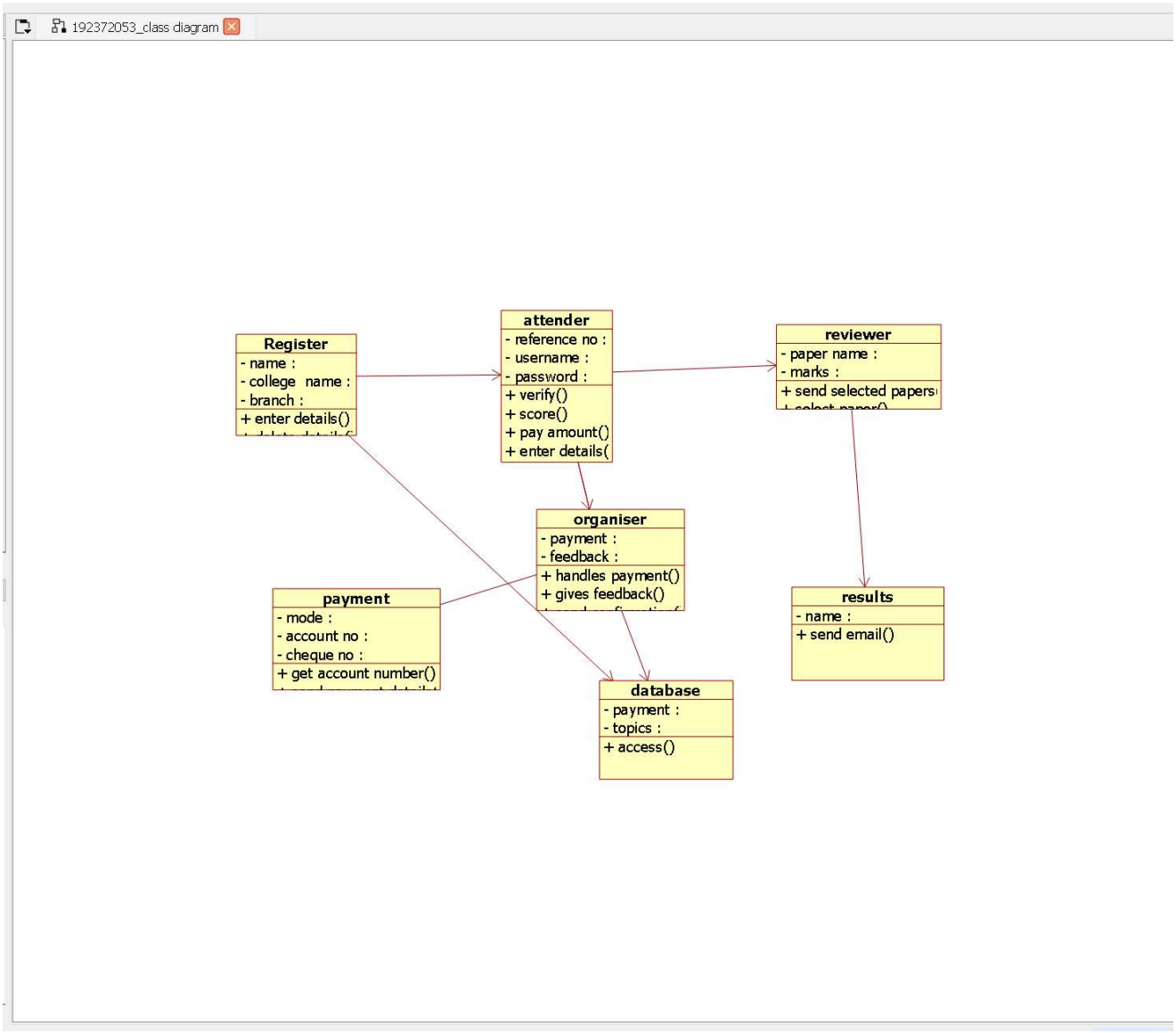
Use case: planning, invite delegates, allocate seats, presenting paper, prize distribution



CLASS DIAGRAM:

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

CLASSES	ATTRIBUTES	OPERATIONS
Member	Name, id	Presenting paper()
Organizer	Member details	Allocating seats()
Central management system	Member details	Updating()



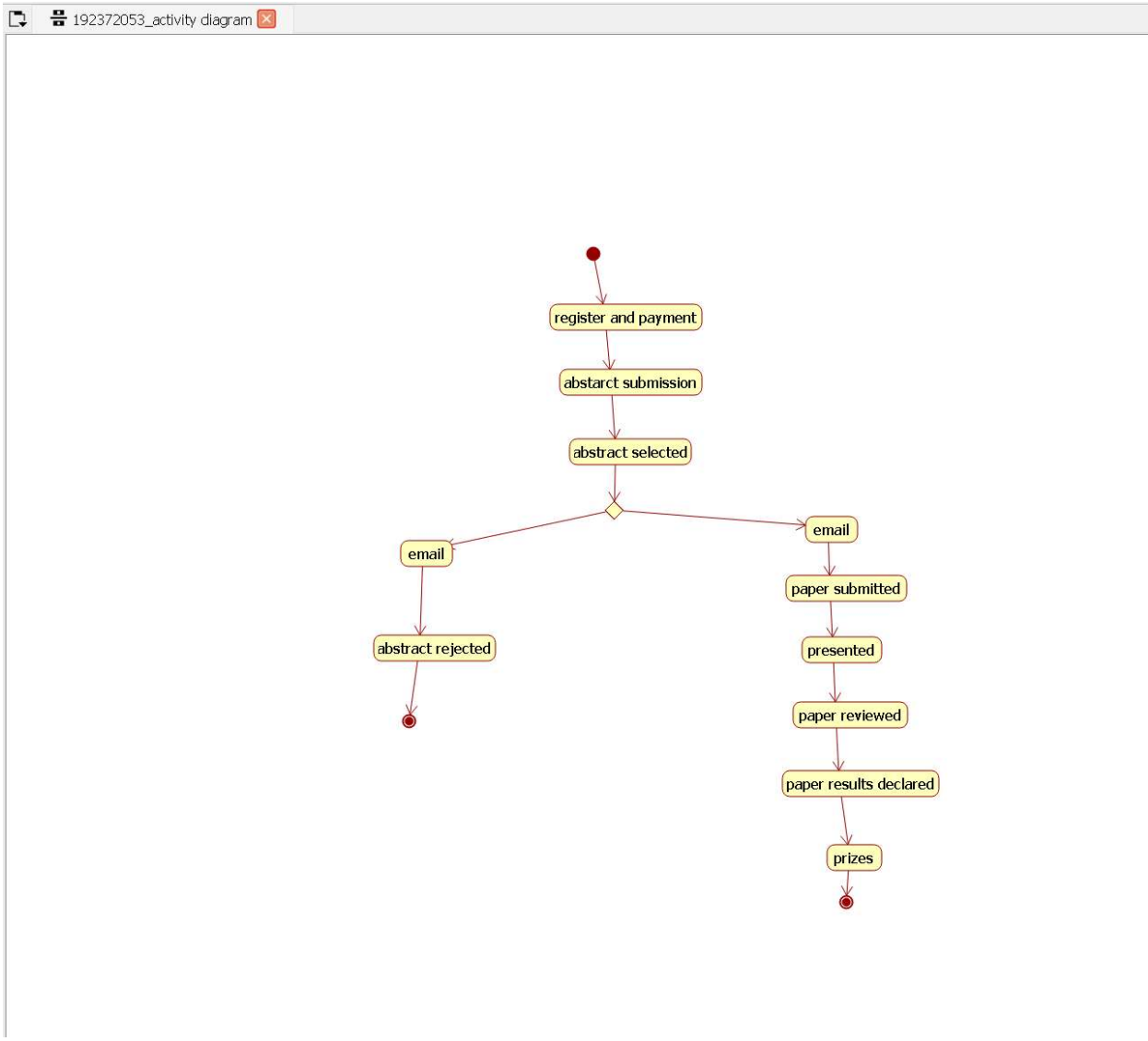
REGISTER NO:

ACTIVITY DIAGRAM:

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

Activities: Invite delegates, Allocate seats, Presenting paper, Choose the winner

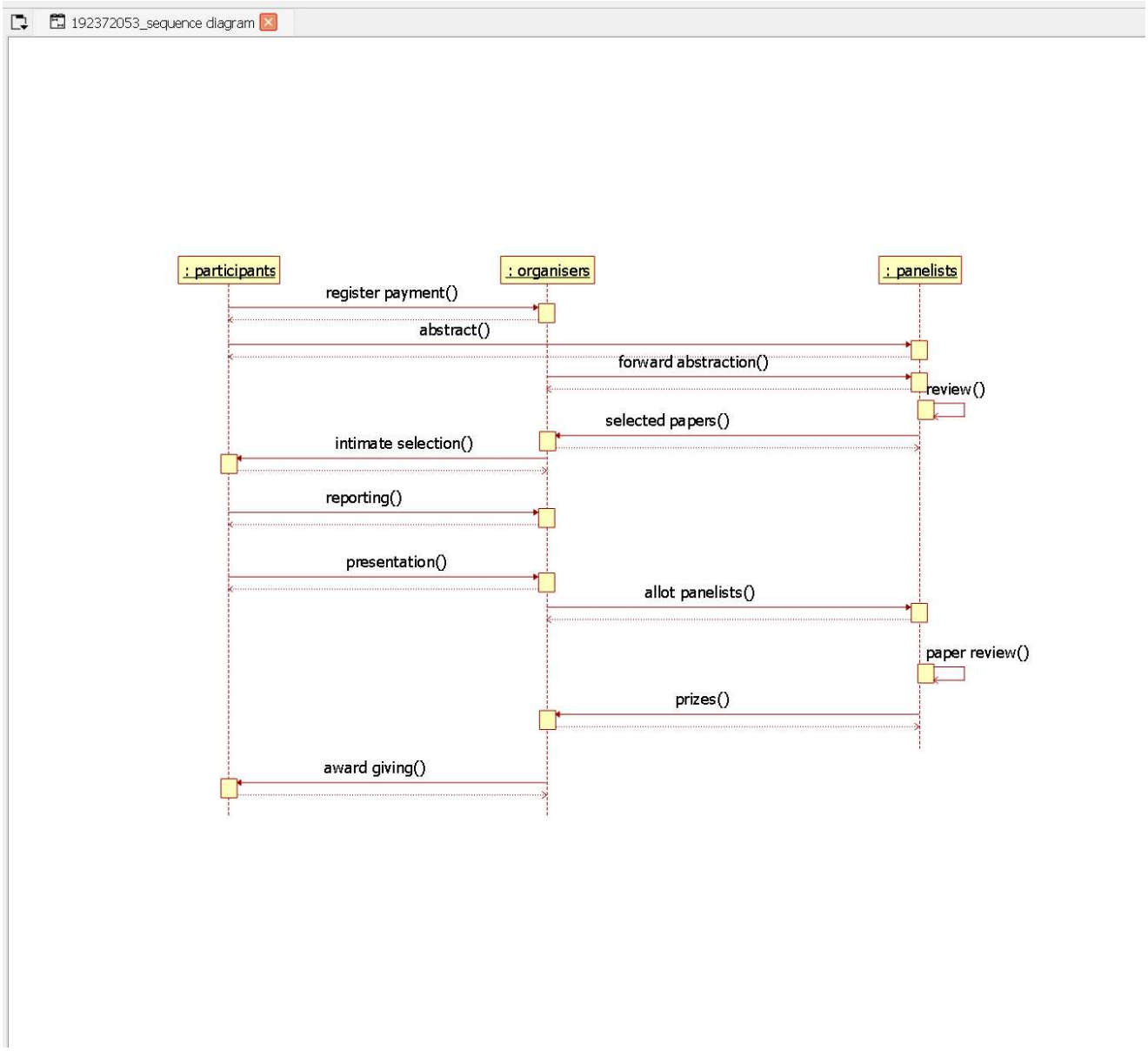
Decision box: Whether it is reserved or not, Whether the presentation is good or not



SEQUENCE DIAGRAM:

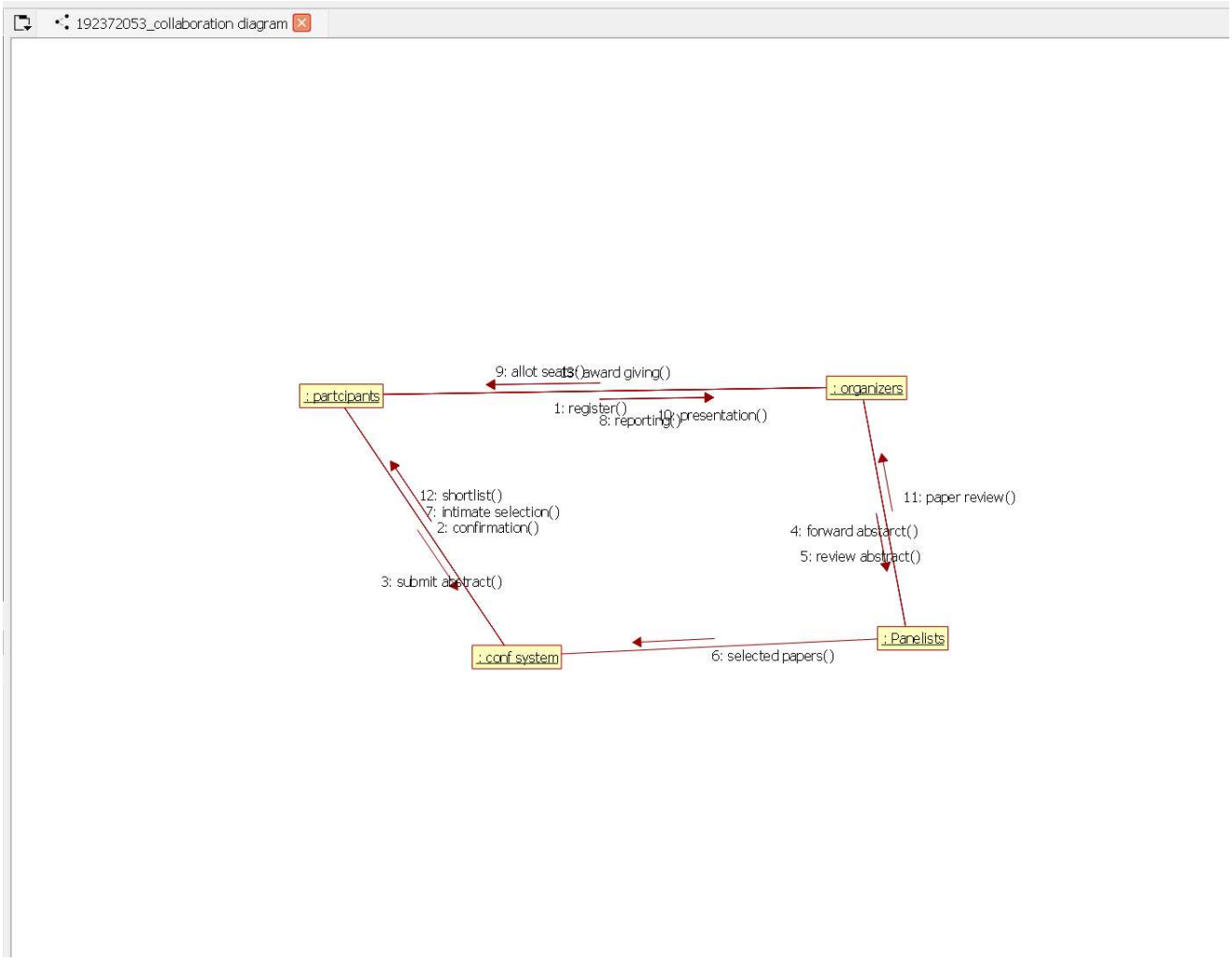
This diagram consists of the objects, messages and return messages.

Object: Member, Organiser, Central management system



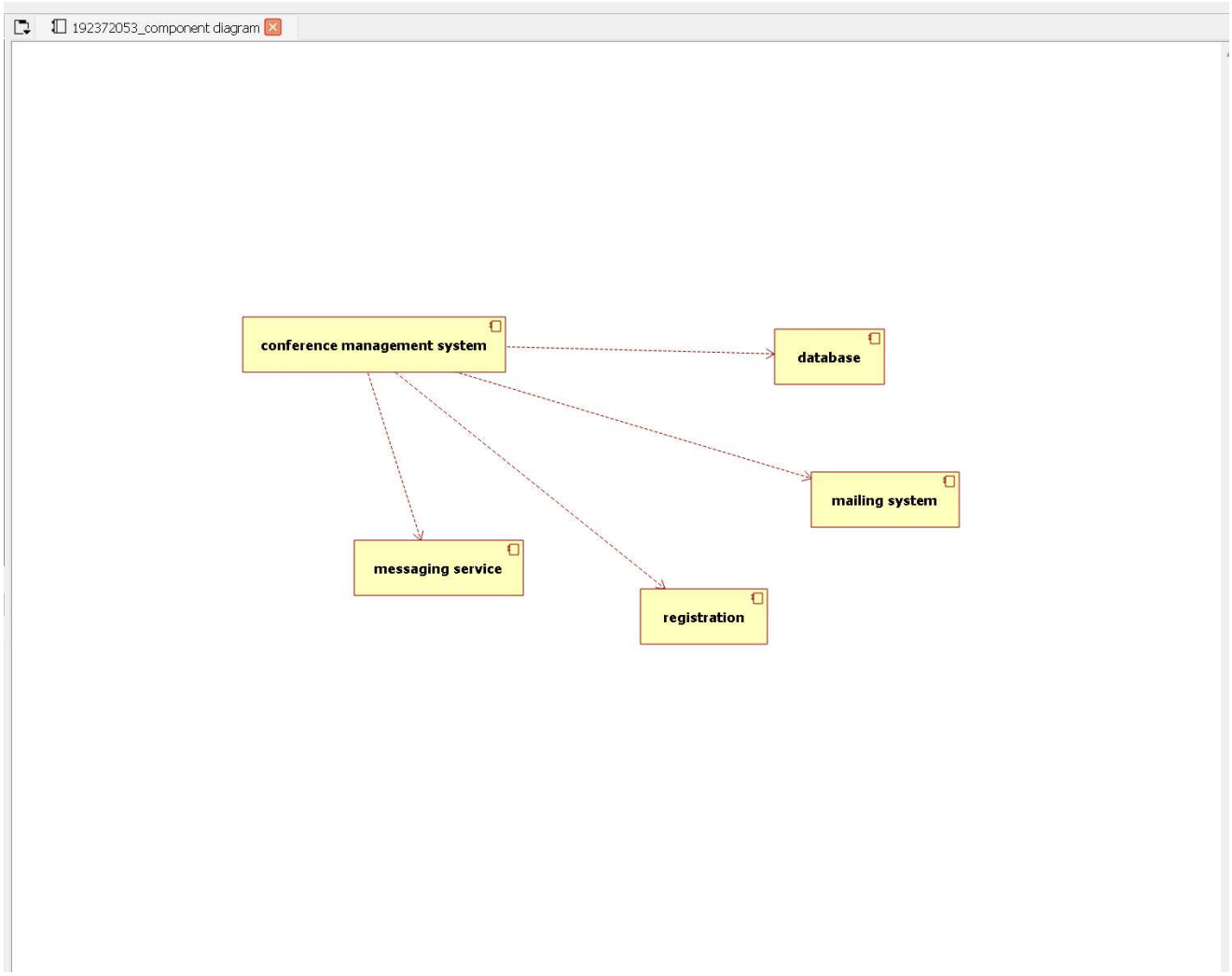
COLLABORATION DIAGRAM:

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



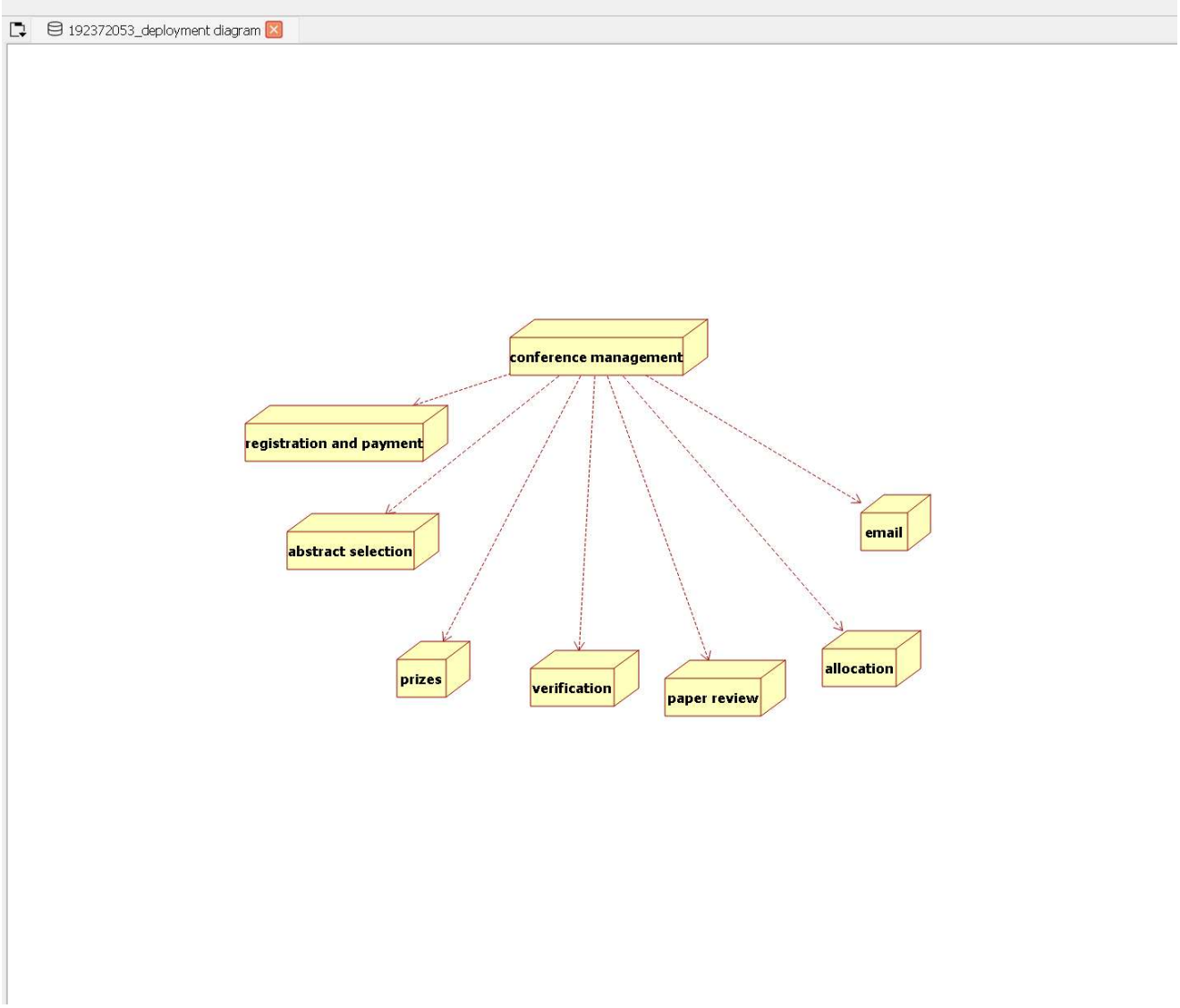
COMPONENT DIAGRAM:

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



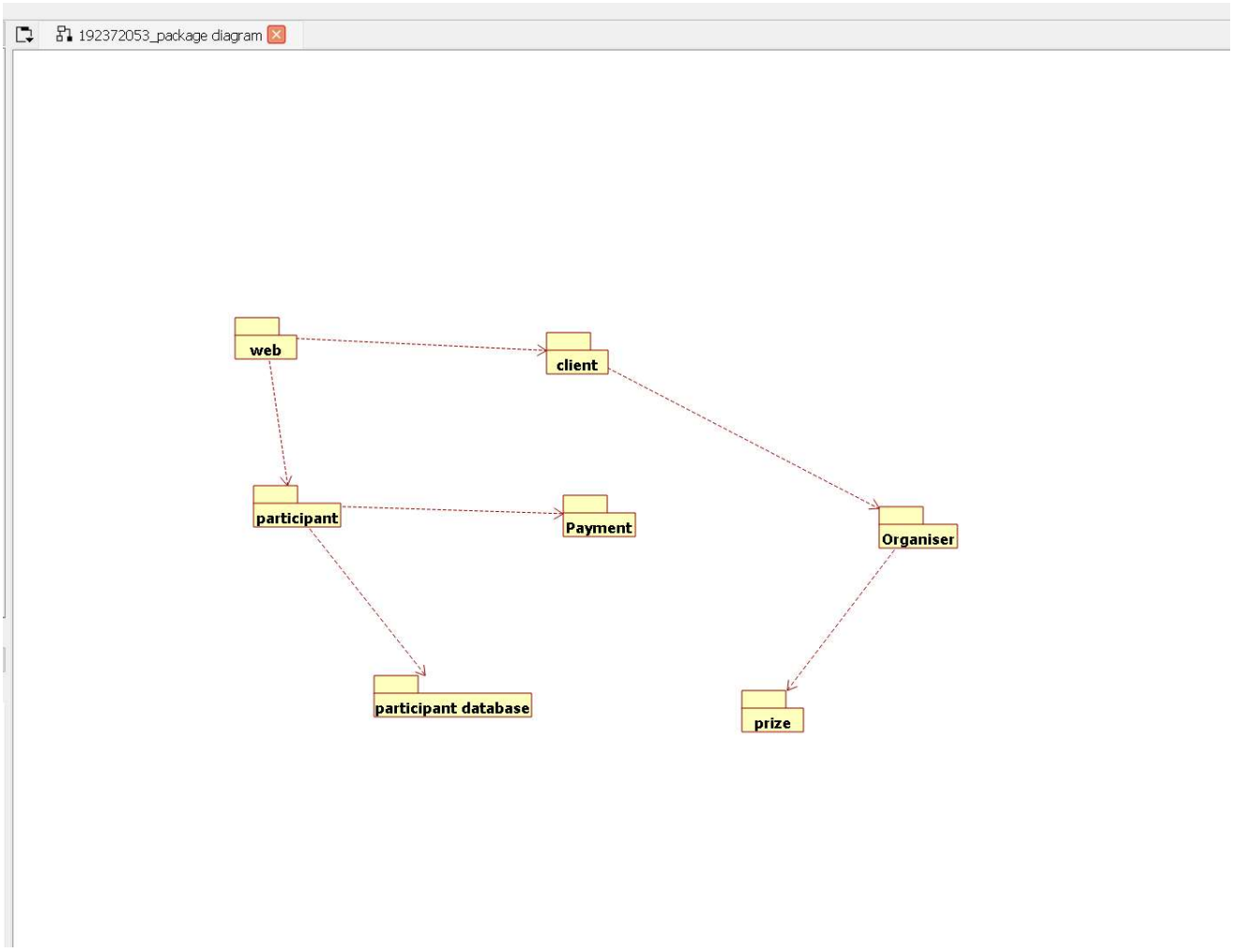
PACKAGE DIAGRAM:

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



REGISTER NO:

PROGRAM CODING:

MEMBER 1:

```
public class member
 {
        public integer name;
        public integer id;
        public integer proof;
        public void winning prize()
         {
         }
        public void member()
         {
```

```
        }
}
```

ORGANIZER:

```
public class organizer
{
        public integer member
        attributes; public integer
        function details;
        public void choosing for
        winner() {
        }
}
```

CENTRAL MANAGEMENT

SYSTEM: public class central
management system {


```
        public integer function details;
        public integer detail of seat allocation;
        public void storing()
        {
        }
        public void updating details()
        {
        }
}
```

RESULT:

Thus draw the diagrams [use case, activity, sequence, collaboration, class, state chart, component, deployment, package] for Conference management system has been designed, executed and output is verified.

DATE:

EX:NO:12

FOREIGN TRADING SYSTEM

AIM:

To draw the diagrams [Use case, Activity, Sequence, Collaboration, Class, State chart, Component, Deployment, package] for foreign trading system

SOFTWARE REQUIREMENTS SPECIFICATION

	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 PROJECT ANALYSIS AND PROJECT PLANNING

The initial requirements to develop the project about the mechanism of the Foreign Trading

System is bought from the trader. The requirements are analyzed and refined which enables the analyst (administrator) to efficiently use the Foreign Trading System. The complete project analysis is developed after the whole project analysis explaining about the scope and the project statement is prepared.

1.3 PROJECT DESCRIPTION:

This software is designed to maintain the details about the trading system that exists between the foreign countries. These details are hold by the trading management system. The details to the system are provided by the customer and the supplier.

1.4 REFERENCES:

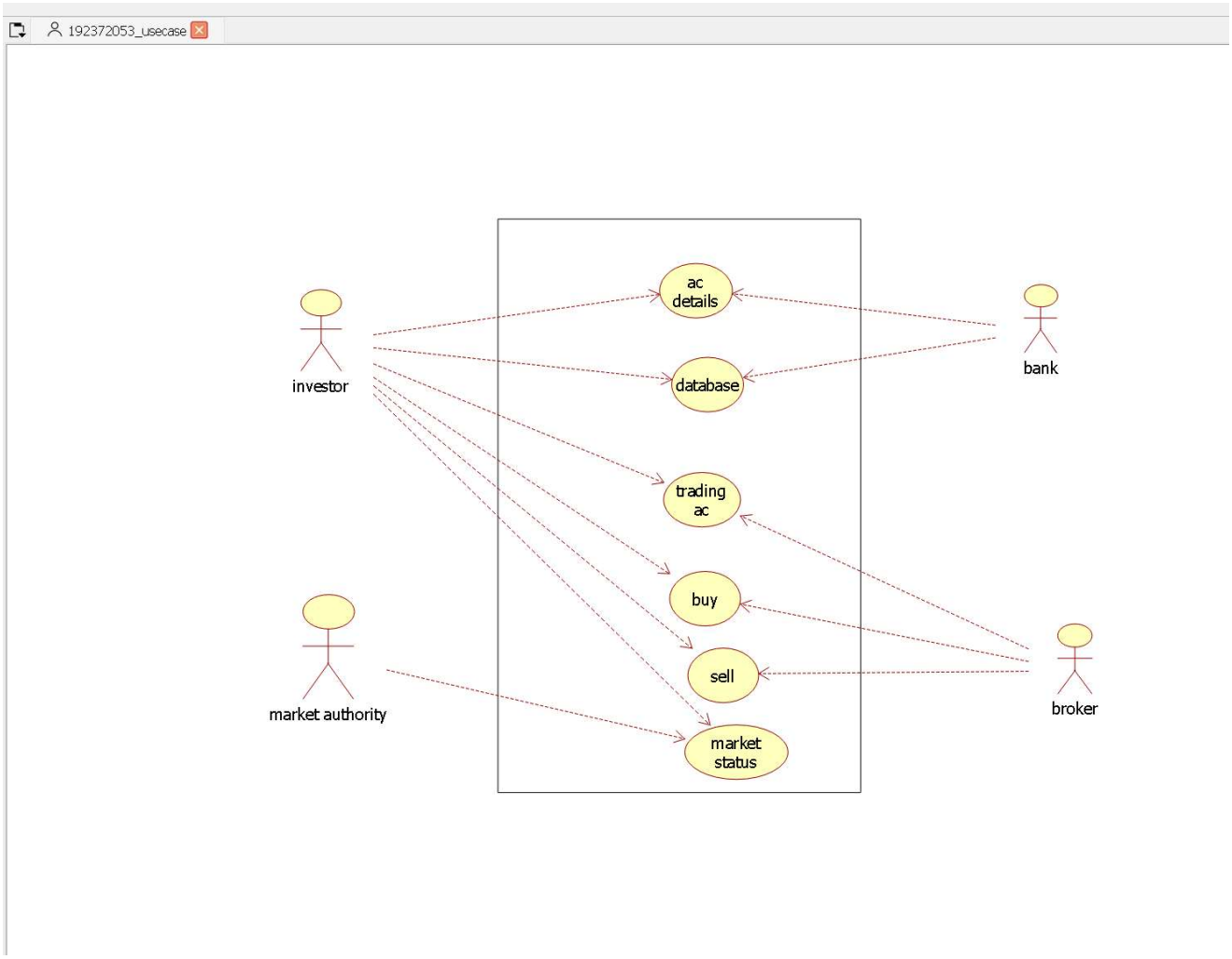
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Customer, Supplier, Custom officer

Use case: Order of product, Quantity, Specify the amount

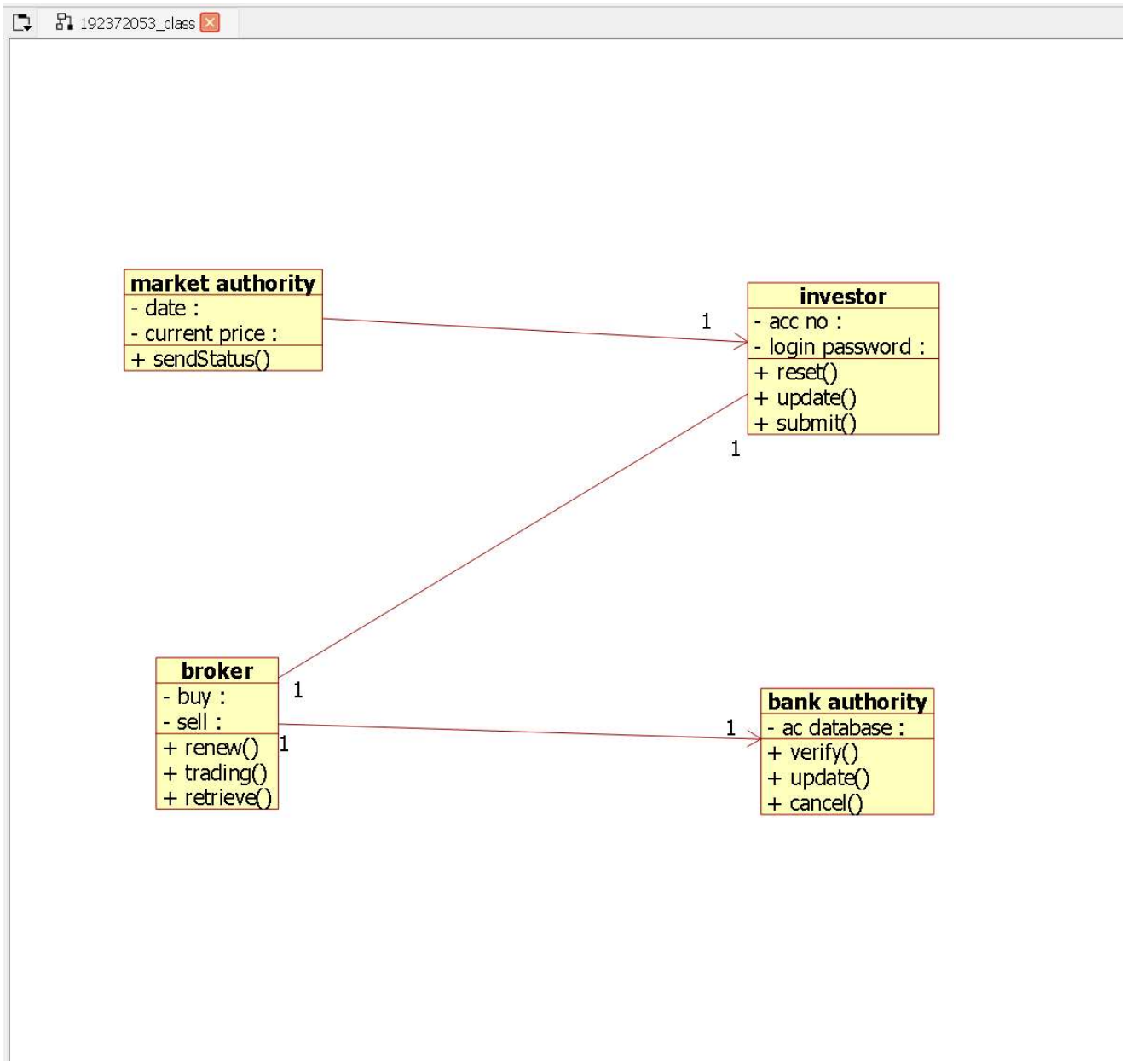


REGISTER NO:

CLASS DIAGRAM:

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

CLASSES	ATTRIBUTES	OPERATIONS
Trading management system	Verify product	Transport()
Customer	Quality	Payment()
Supplier	Product supply	Money transfer()

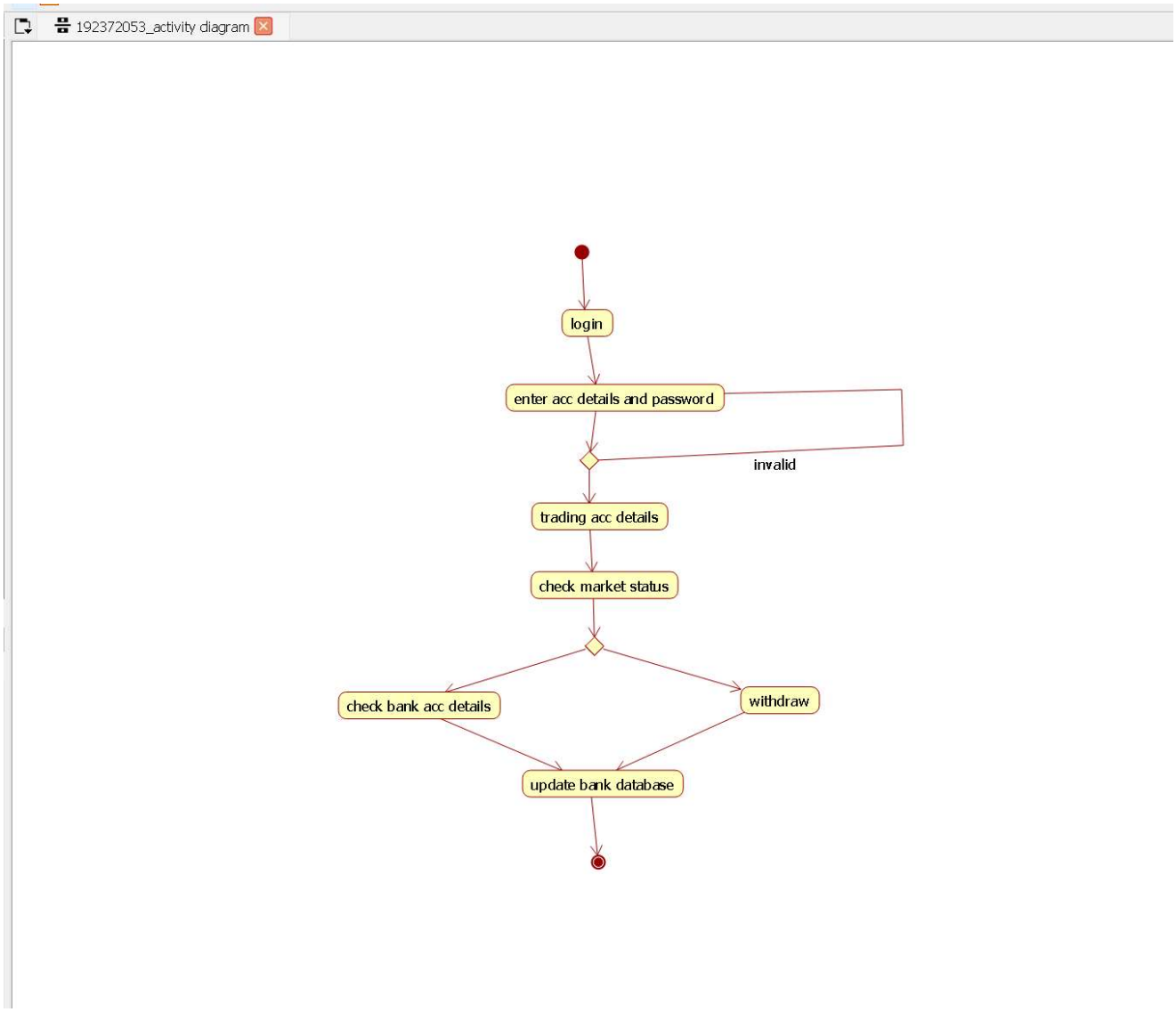


ACTIVITY DIAGRAM:

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

Activities: Order of the product, Specify amount, Payment, Money transfer

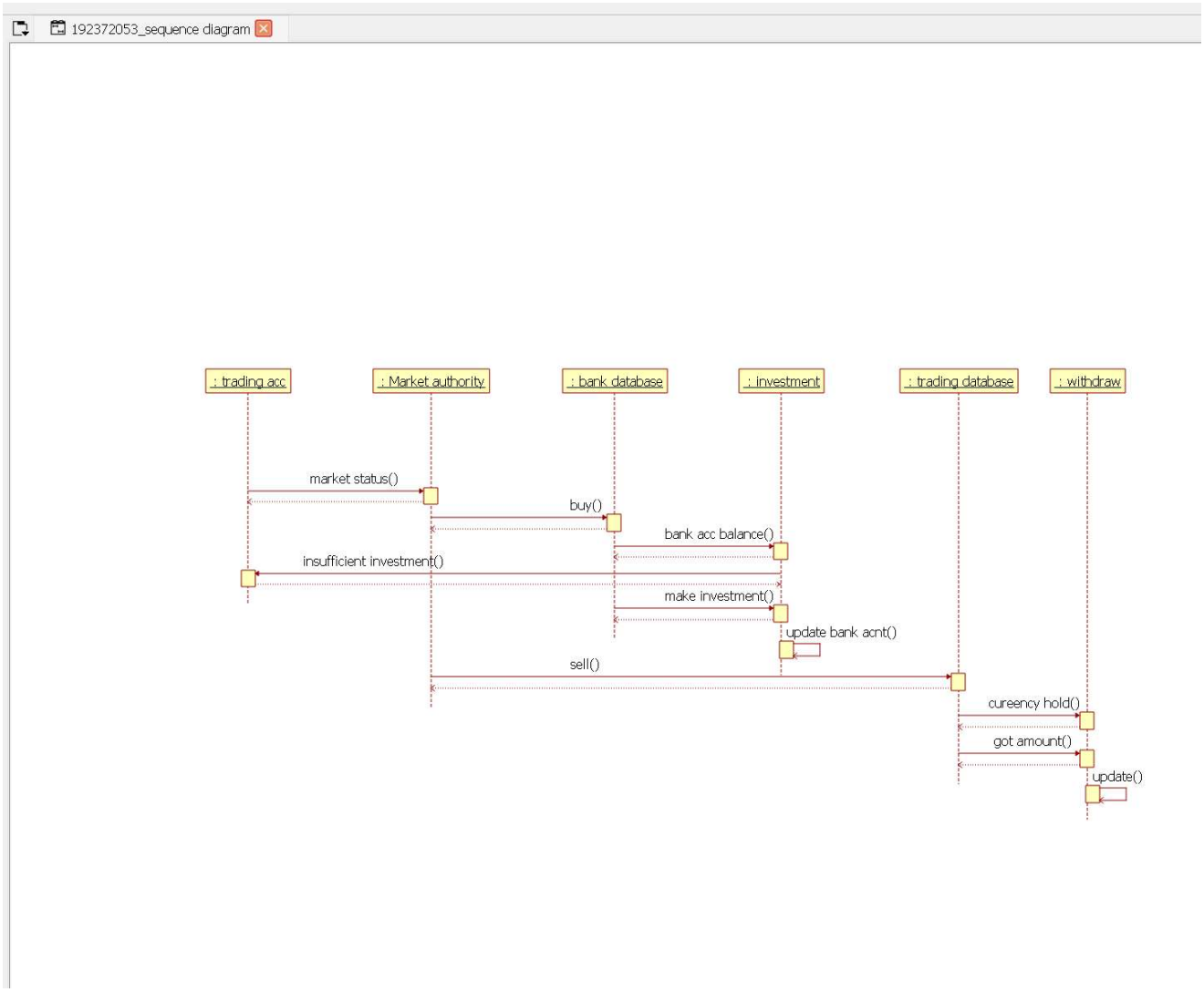
Decision box: Check for availability



SEQUENCE DIAGRAM:

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

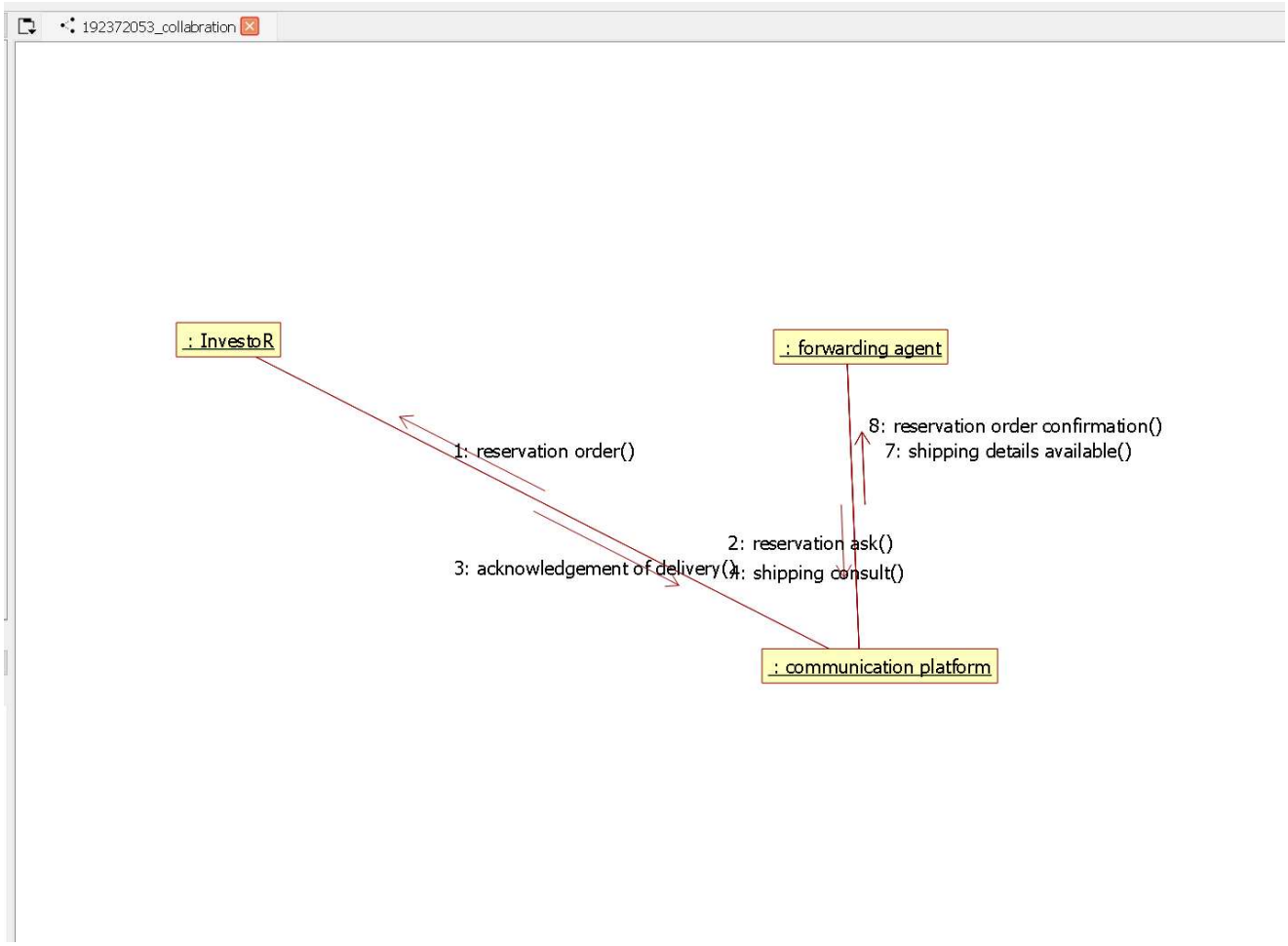
Object: Customer, Supplier, Trading management system



REGISTER NO:

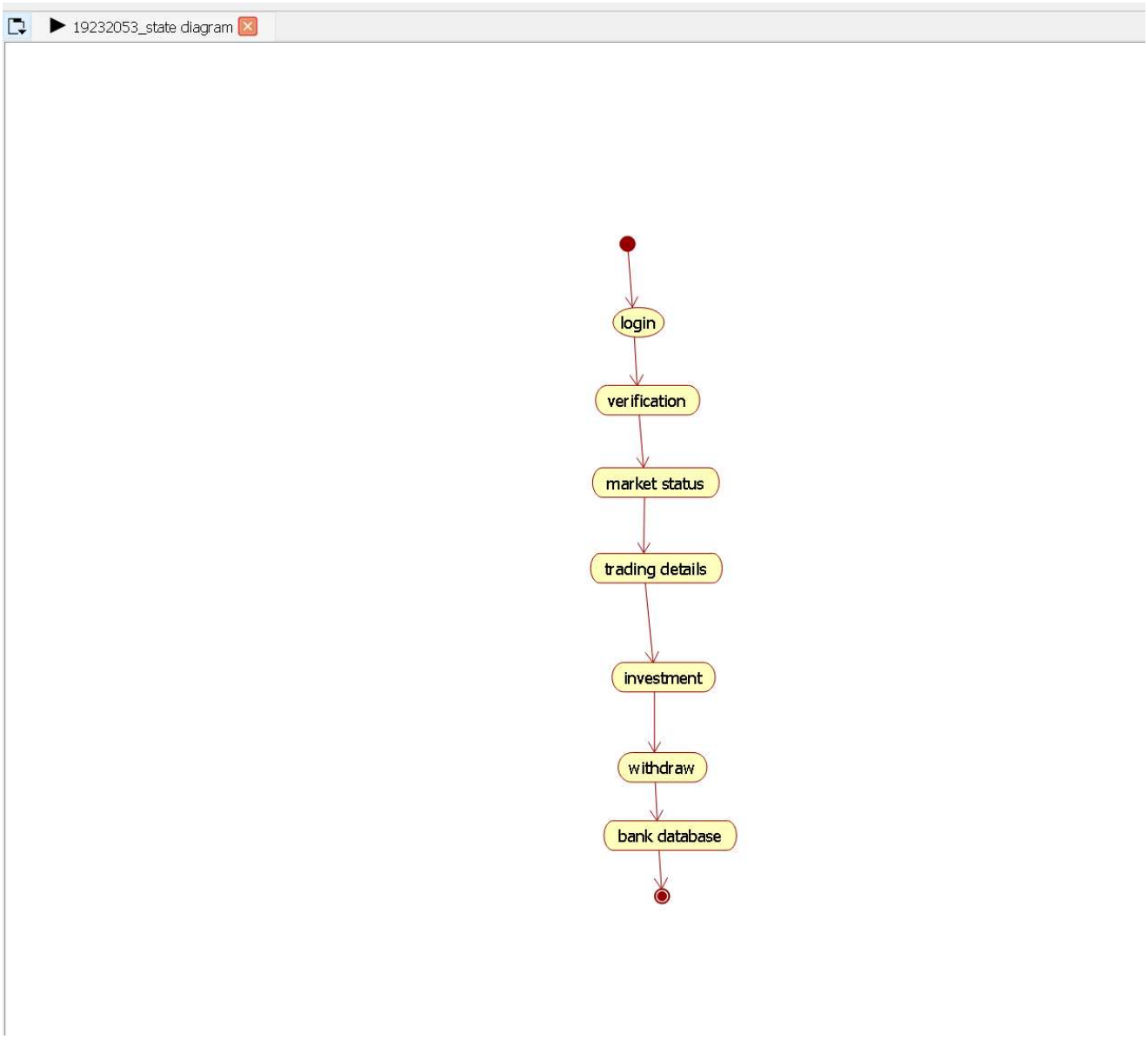
COLLABORATION DIAGRAM:

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



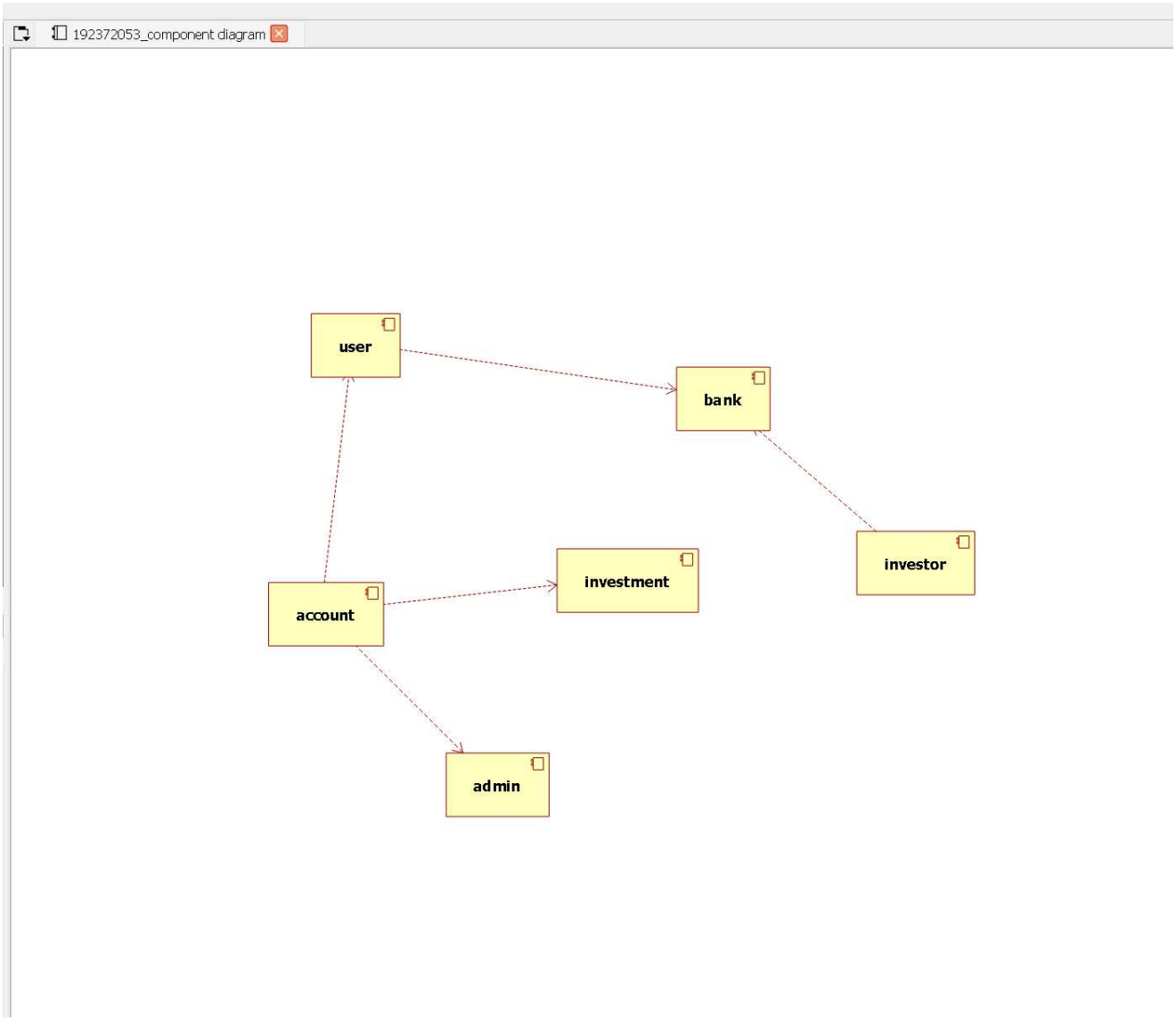
STATECHART DIAGRAM:

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:

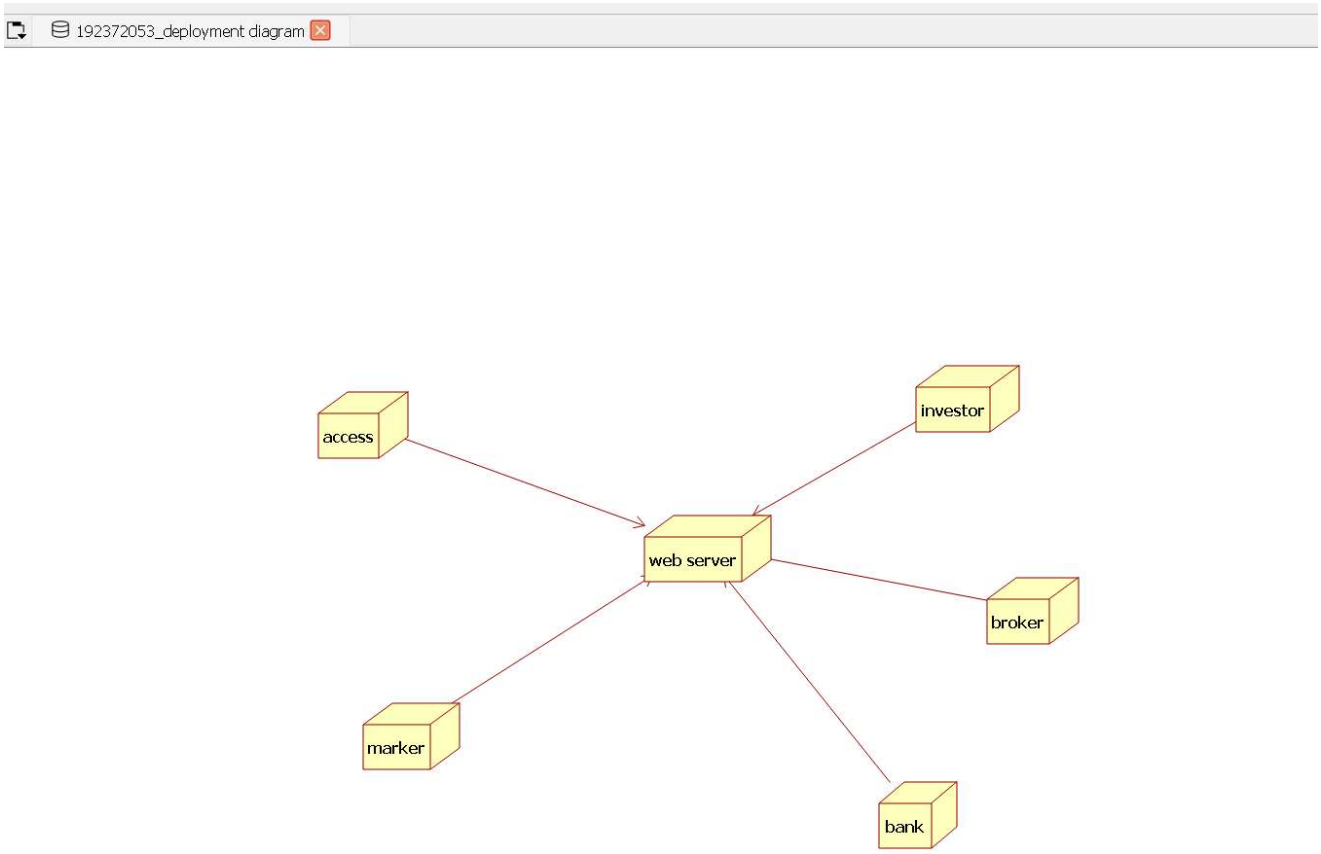
The component diagram is represented by figure dependency and it is a graph of design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a systems. It is represented by boxed figure. Dependencies are represented by communication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



REGISTER NO:

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

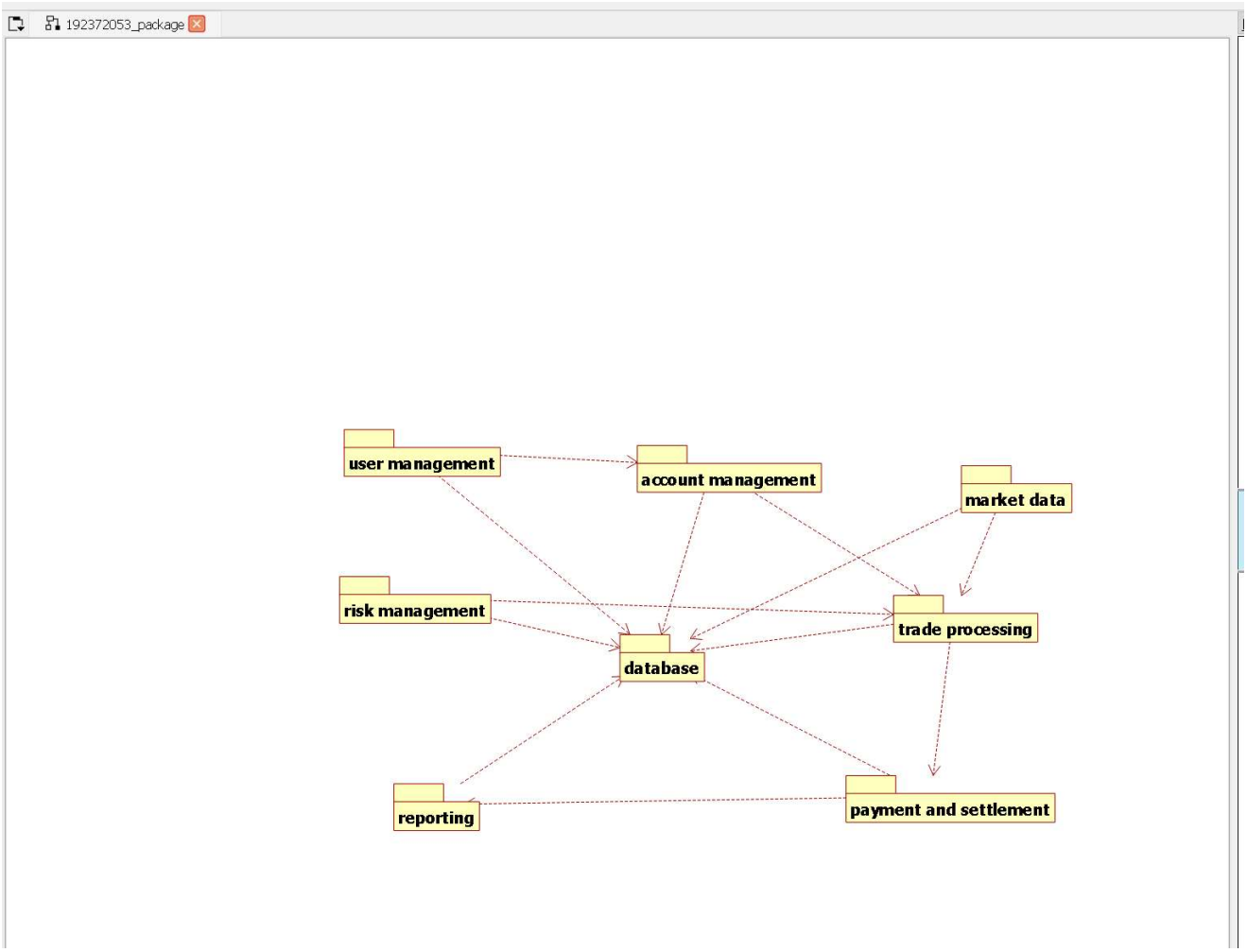


PACKAGE DIAGRAM:

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:

TRADING MANAGEMENT SYSTEM:

```
public class trading management system
 {
        public integer verify product;
        public integer amount;
        public void transport()
         {
         }
        public void money transfer()
         {
         }
```

}

CUSTOMER:

```
public class customer
{
        Public integer order product;
        Public integer amount;
        Public void payment()
         {
         }
        Public void delivery()
         {
         }
}
```

SUPPLIER:

```
Public class supplier
{
        Public integer supply;
        Public void available product()
         }
```

RESULT:

Thus thethe diagrams [Usecase, Activity, Sequence, Collaboration, Class, Statechart, Component, Deployment, package] for foreign trading system has been designed, executed and output is verified.

Ex.no 13	**BPO MANAGEMENT SYSTEM**
Date:	

AIM:

To draw the diagrams [Use case , Class, Activity, Sequence, Collaboration, State Chart, Component, Deployment, package] for the BPO Management System .

OOAD LAB

REGISTER NO:

SOFTWARE REQUIREMENTS SPECIFICATION

	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

1.3 PROJECT DESCRIPTION:

This software is designed to know about the process that were taking place in the BPO office. This system holds the details of the customer who and all approaches to it.It is managed by the central systems.

1.4 REFERENCES:

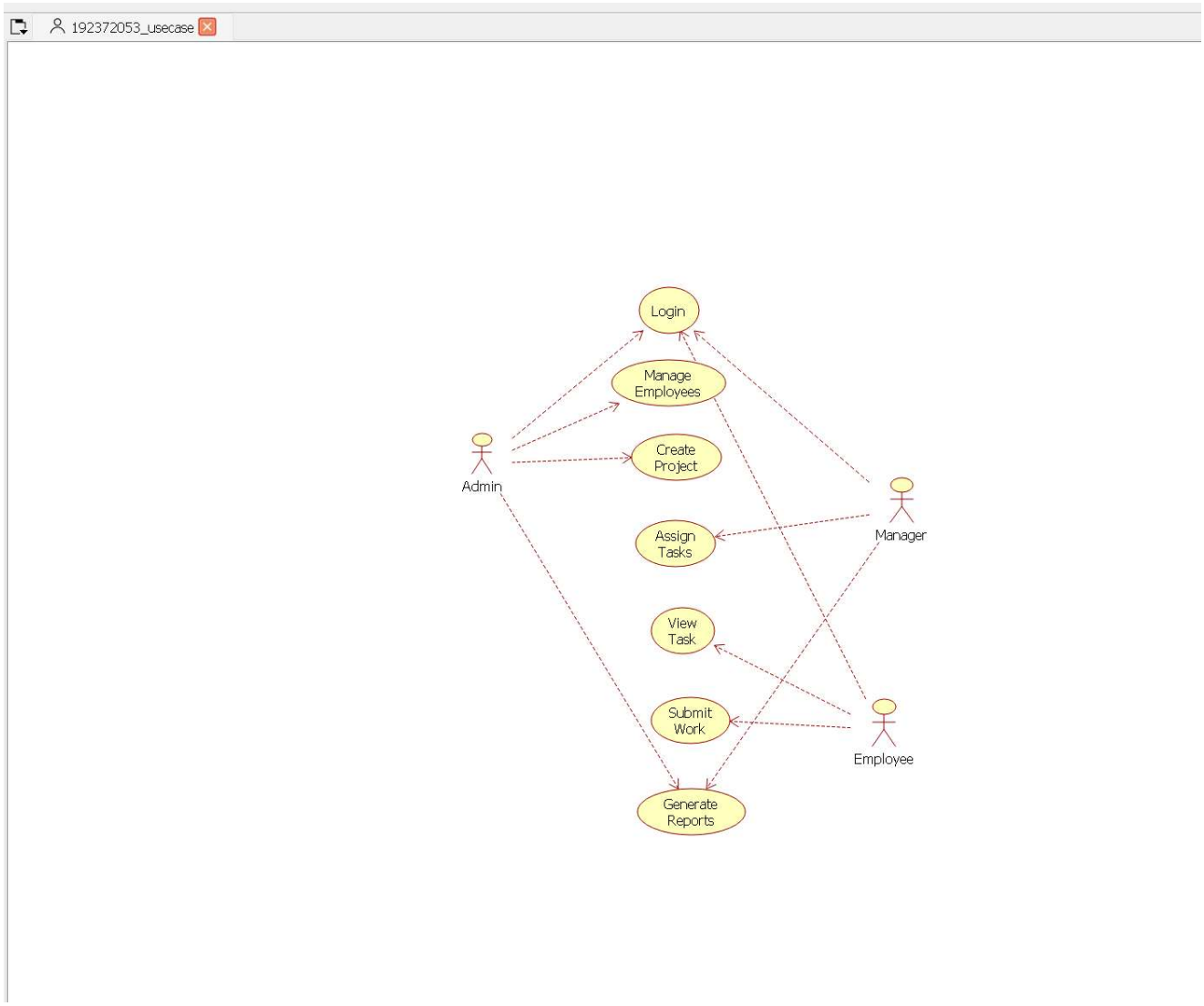
IEEE Software Requirement Specification format.

USECASE DIAGRAM:

This diagram will contain the attributes as start point, end point, decision box as given below

ACTORS: Purchase product, Server, Central system

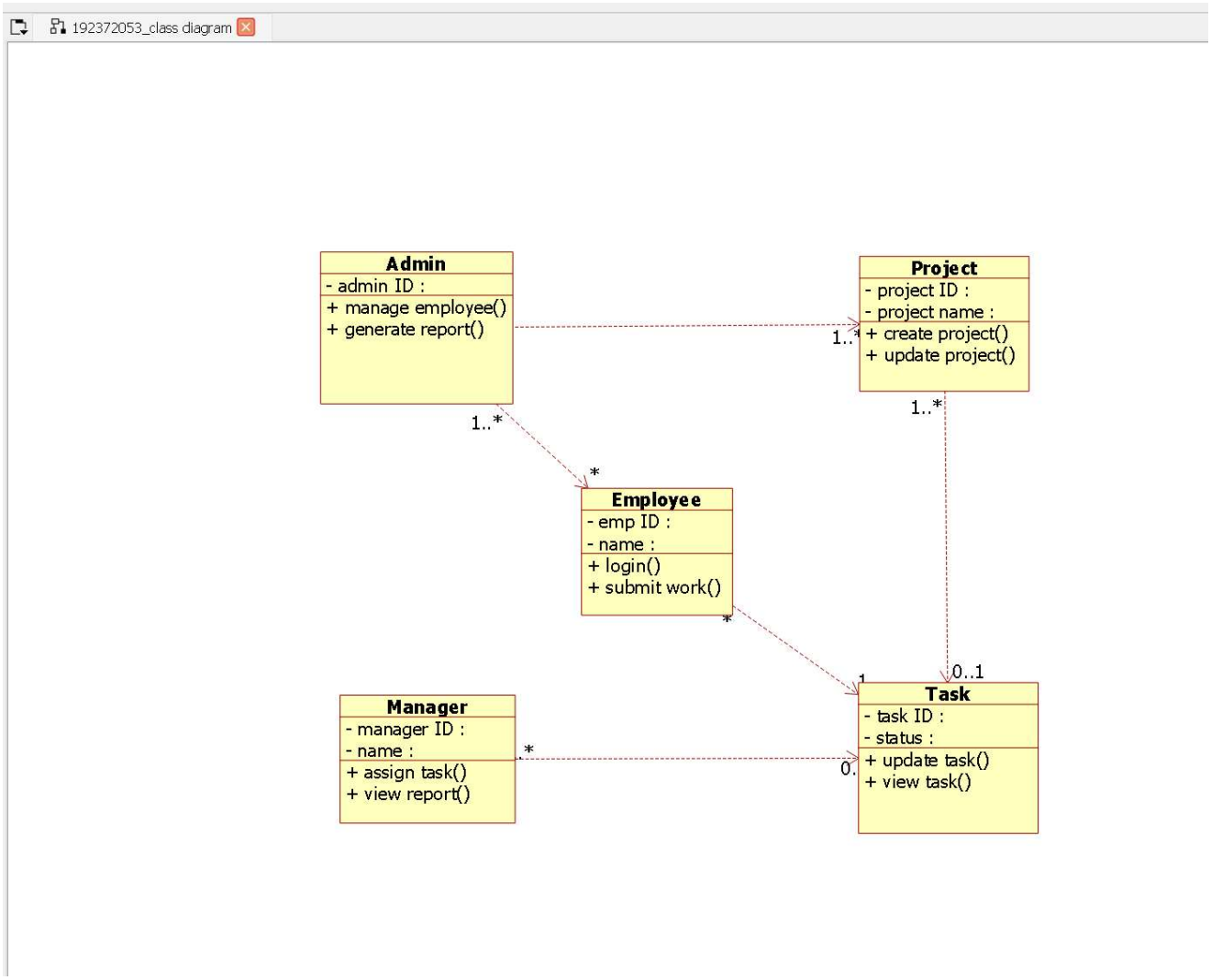
USECASE: Product, Voice, Non-Voice, Indian office, Employee, Feedback.



CLASS DIAGRAM:

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

CLASSES	ATTRIBUTES	OPERATIONS
Central System	Store, update	Storing(), updating()
Dealer	Employee name	Delivery()
Customer	Details	Feedback()



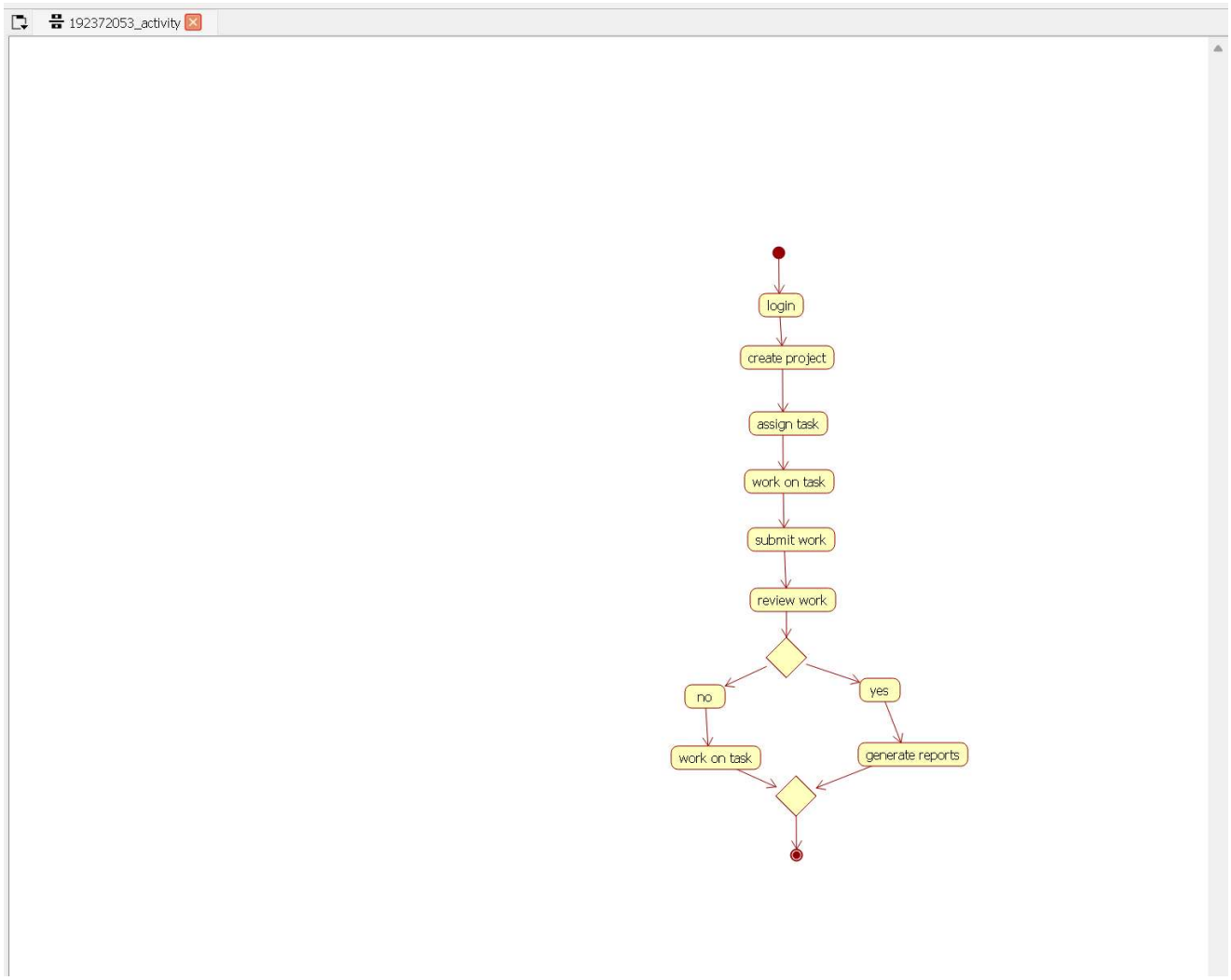
REGISTER NO:

ACTIVITY DIAGRAM:

This diagram will contain the activities as start point, end point, decision boxes as given below

ACTIVITIES: Purchase Product, On call, On chat

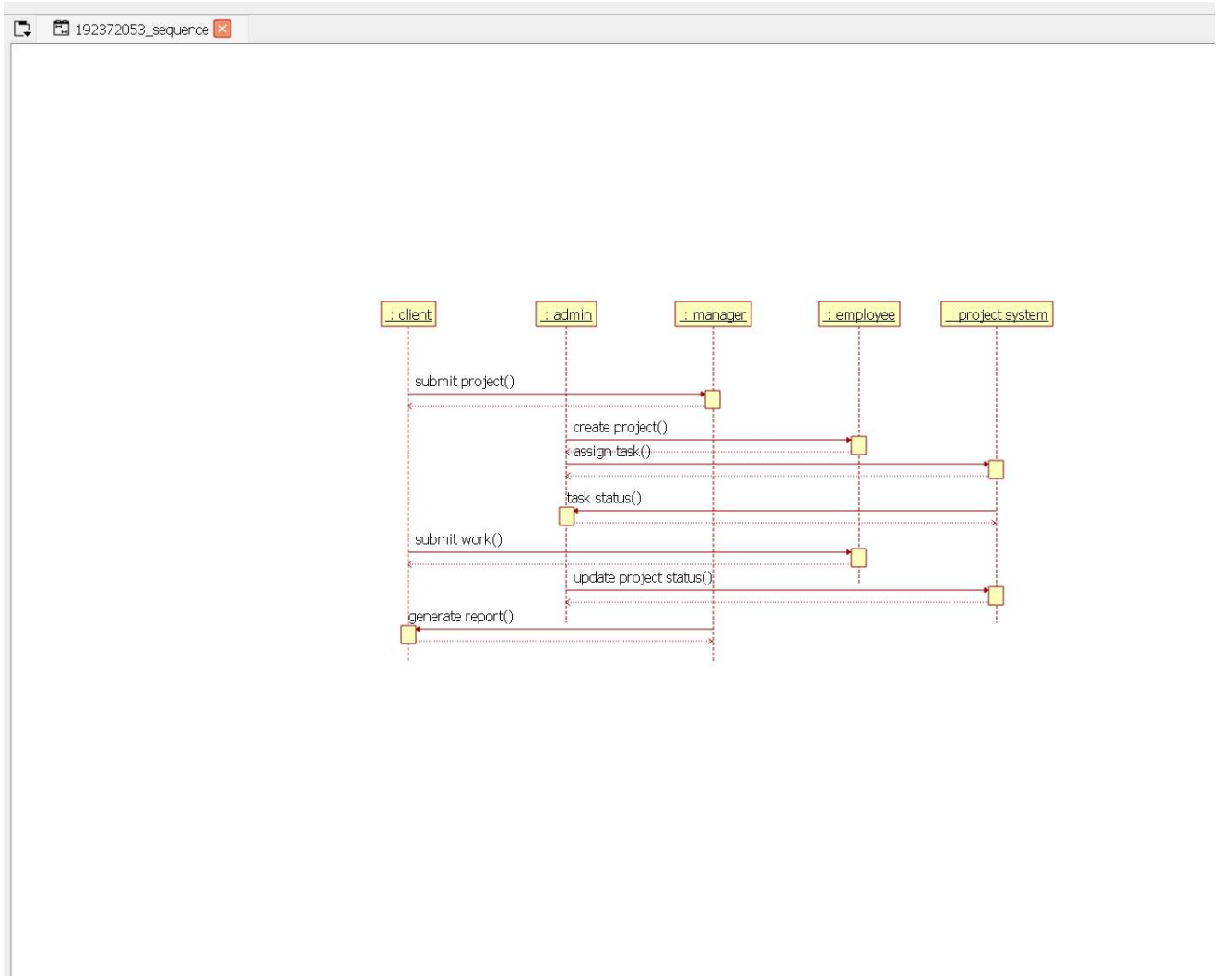
DECISION BOX: Option to check



SEQUENCE DIAGRAM:

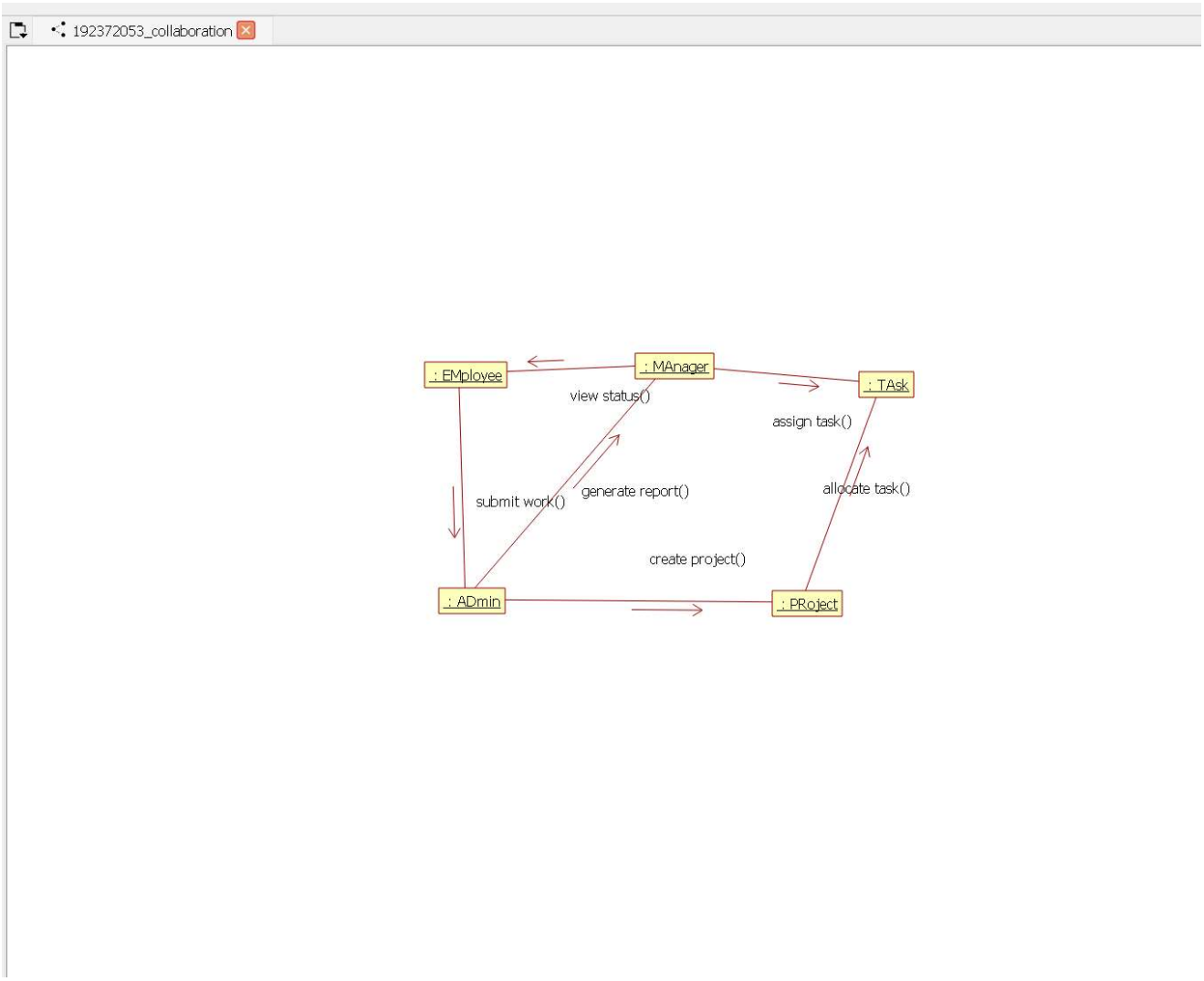
This diagram consists of the objects, messages and return messages

Object: Customer, Dealer, Central System



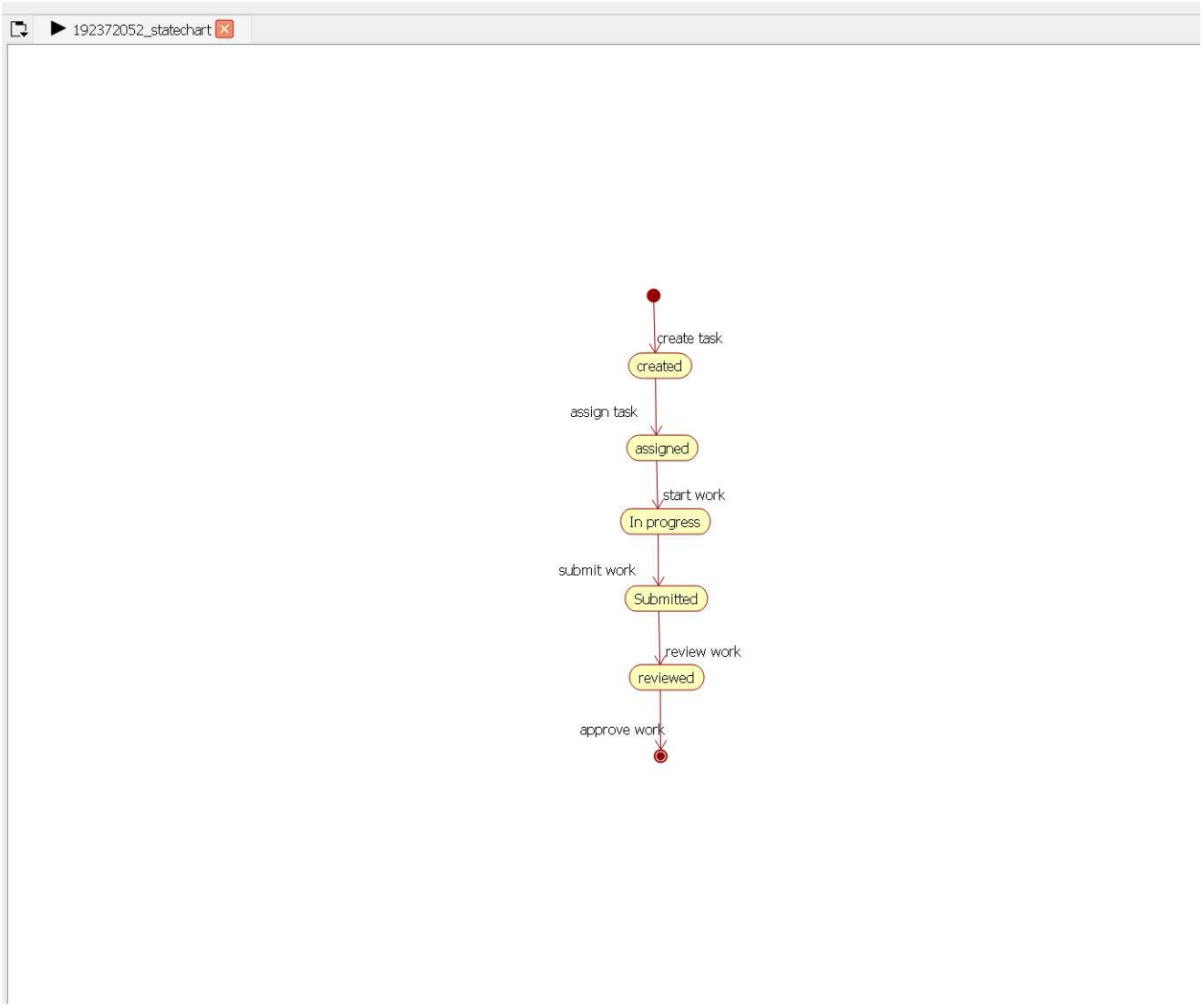
COLLABORATION DIAGRAM:

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



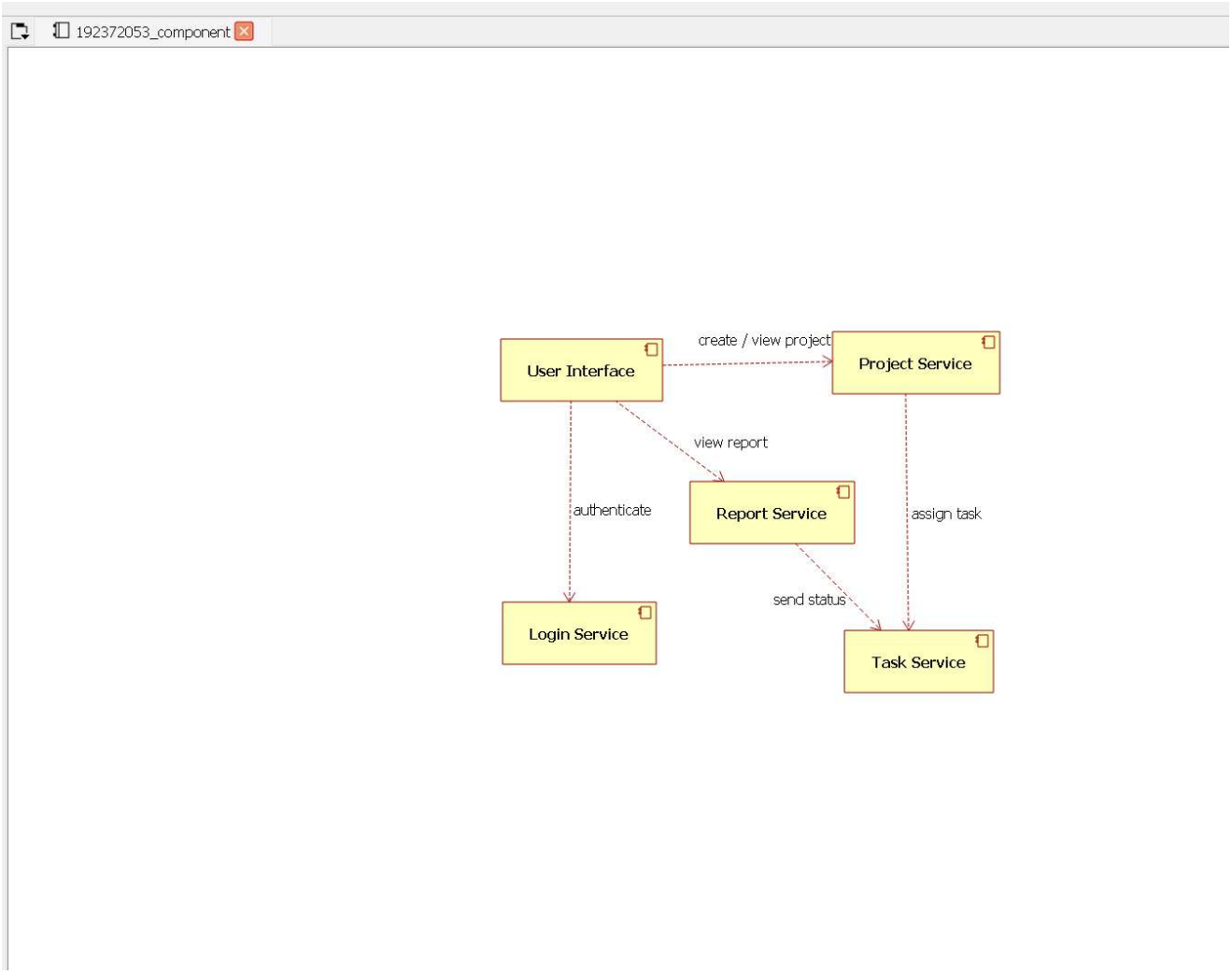
STATECHART DIAGRAM:

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



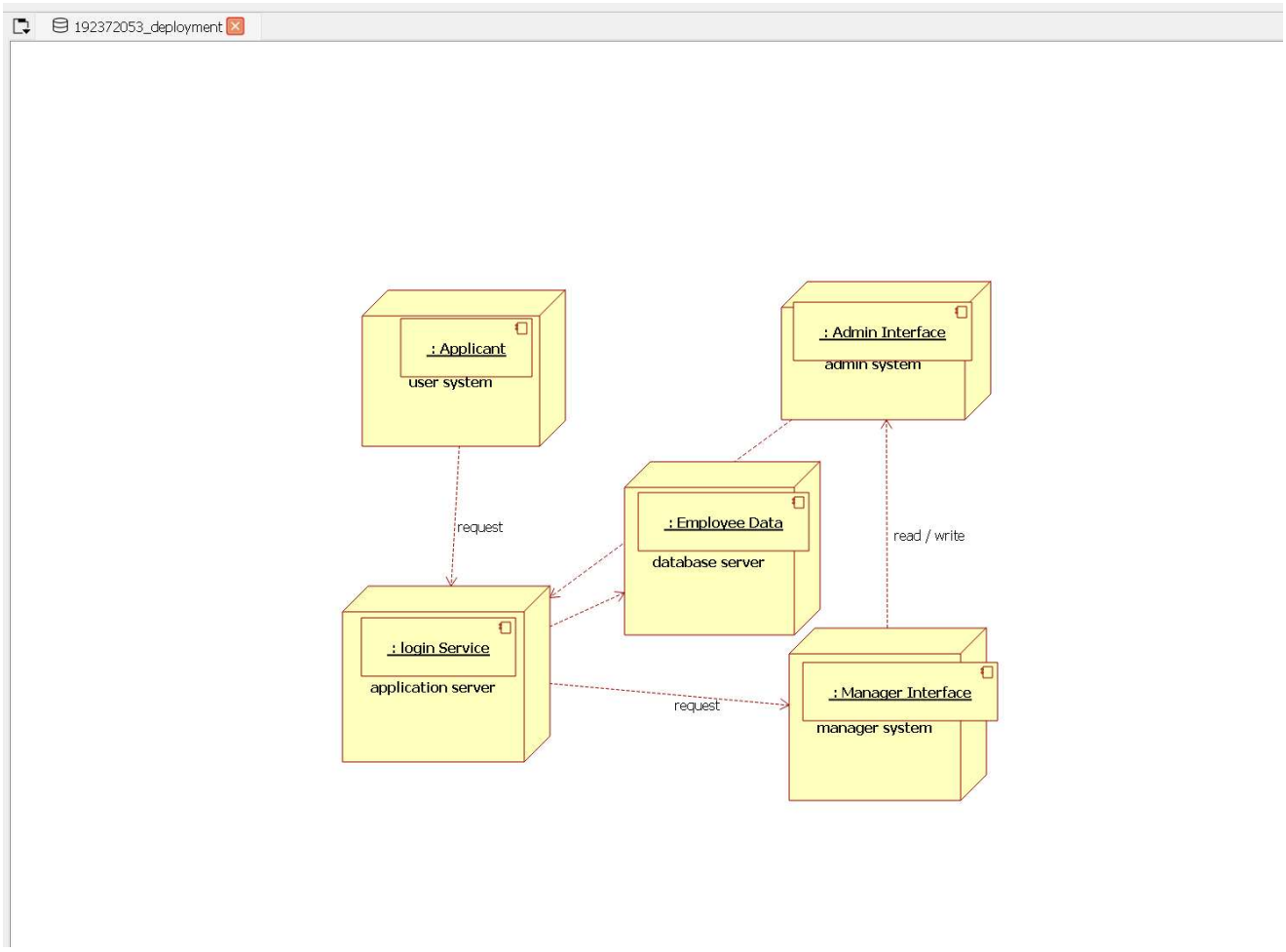
COMPONENT DIAGRAM:

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 associatio



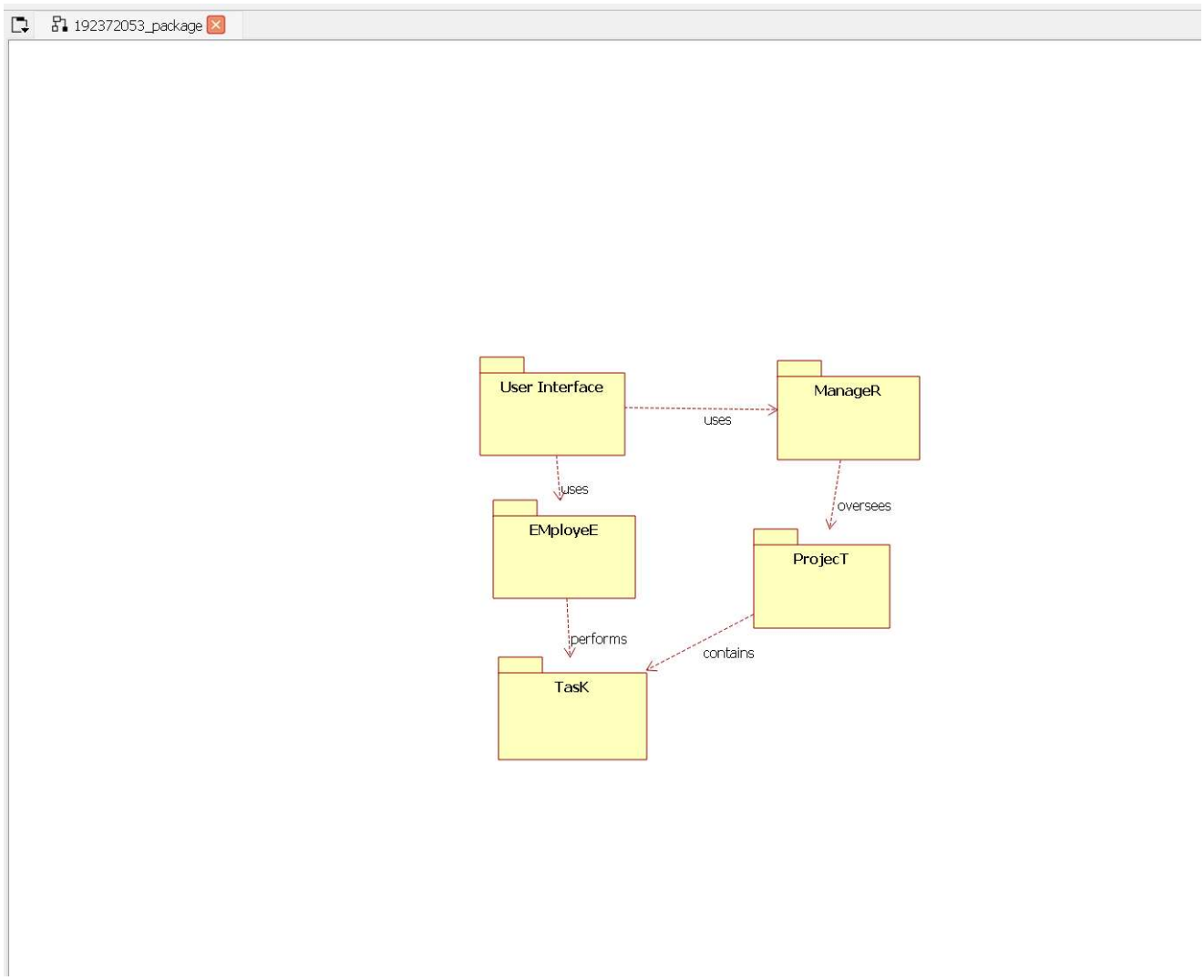
REGISTER NO:

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:

CENTRAL SYSTEM:

```
import java.util.Vector;
public class central system
  {
        public Integer store;
         public Integer update;
         public Vector mydealer;
         public void updating()
         {
         }
```

```
        public void processing()
          {
          }
}
```


CUSTOMER:

```
import java.util.Vector;
public class customer
 {
        public Integer name;
        private Integer product;
        public Vector mydealer;
         public void feedback()
         {
          }
         public void customer()
         {
          }
}
```

DEALER:

```
import java.util.Vector;
public class dealer
 {
        public Integer employeename;
        public Integer availability;
```

```
public Integer newAttr;
 public Vector mycustomer;
 public Vector mycentral system;
public void payment()
  {
 }
 public void delivery()
```



```
  {
 }
}
```

RESULT:

The diagrams [Use case, Class, Activity, Sequence, Collaboration, State Chart, Component, Deployment, package] for the BPO Management system has been designed, executed and output is verified.

OOAD LAB