Ex no:

EXAM REGISTRATION SYSTEM

Date:

AIM:

To create a system to perform the Exam Registration system

PROBLEM STATEMENT:

Exam Registration system.is used in the effective dispatch of registration form to all of the students. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner. The core of the system is to get the online registration form (with details such as name, reg.no etc.,) filled by the student whose testament is verified for its genuineness by the Exam Registration System with respect to the already existing information in the database. This forms the first and foremost step in the processing of exam application. After the first round of verification done by the system, the information is in turn forwarded to the Exam Controller. The application is then processed manually based on the report given by the system. The system also provides the student the list of exam dates.The controller will be provided with fees details to display the current status of application to the student, which they can view in their online interface. After all the necessary criteria has been met, the original information is added to the database and the hall ticket is sent to the student.

(I)SOFTWARE REQUIREMENT SPECIFICATION:

1.0 INTRODUCTION

Exam Registration System is an interface between the Student and the Exam Controller responsible for the Issue of Hall Ticket. It aims at improving the efficiency in the Issue of Hall ticket and reduces the complexities involved in it to the maximum possible extent.

1.1 PURPOSE

If the entire process of 'Issue of Hall ticket' is done in a manual manner then it would takes several days for the hall ticket to reach the student. Considering the fact that the number of students for hall ticket is increasing every year, an Automated System becomes essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process. As this is a matter of National Security, the system has been carefully verified and validated in order to satisfy it.

1.2 SCOPE

• The System provides an online interface to the user where they can fill in their personal details and submit the necessary documents (may be by scanning).

• The controller concerned with the issue of hall ticket can use this system to reduce his workload and process the application in a speedy manner.

• Provide a communication platform between the student and the controller.

• Students will come to know their status of application and the date in which they must subject themselves for manual document verification.

1.3 DEFINITIONS, ACRONYMS AND THE ABBREVIATIONS
• **Exam Controller** - Refers to the super user who is the Central Authority who has been vested with the privilege to manage the entire system.
• **Student** - One who wishes to obtain the Hall Ticket.
• **ERS** - Refers to this Examination Registration System.
• **HTML** - Markup Language used for creating web pages.
• **J2EE** – Java 2 Enterprise Edition is a programming platform java platform for developing and   running distributed java applications.
• **HTTP** - Hyper Text Transfer Protocol.
• **TCP/IP** – Transmission Control Protocol/Internet Protocol is the communication protocol used to connect hosts on the Internet.

1.4 REFERENCES
IEEE Software Requirement Specification format.

1.5 TECHNOLOGIES TO BE USED
• HTML
• JSP
• Javascript
• Java
1.6 TOOLS TO BE USED
• Eclipse IDE (Integrated Development Environment)
• Rational Rose tool (for developing UML Patterns)

1.7 OVERVIEW
SRS includes two sections overall description and specific requirements - **Overall Description** will describe major role of the system components and inter-connections. **Specific Requirements** will describe roles & functions of the actors.

2.0.OVERALL DESCRIPTION
2.1 PRODUCT PERSPECTIVE
The ERS acts as an interface between the 'student' and the 'exam controller'. This system tries to make the interface as simple as possible and at the same time not risking the security of data stored in. This minimizes the time duration in which the user receives the hall ticket.

2.2 SOFTWARE INTERFACE

- **Front End Client** - The exporter online interface is built using JSP and HTML.
- **Web Server** – Apache Tomcat Server (Oracle Corporation)
- **Back End** - Oracle 11g database

2.3 HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have access to the database in the server.

2.4 SYSTEM FUNCTIONS

- Secure Registration of information by the Students.
- SMS and Mail updates to the students by the controller.
- Controller can generate reports from the information and is the only authorized personnel to add the eligible application information to the database.

2.5 USER CHARACTERISTICS

- **Student** - They are the people who desire to obtain the hall ticket and submit the information to the database.
- **Exam controller** - He has the certain privileges to add the registration status and to approve the issue of hall ticket. He may contain a group of persons under him to verify the documents and give suggestion whether or not to approve the dispatch of hall ticket.

2.6 CONSTRAINTS

- The applicants require a computer to submit their information.
- Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring.
- The user has to be careful while submitting the information. Much care is required.

2.7 ASSUMPTIONS AND DEPENDENCIES

- The Students and Exam Controller must have basic knowledge of computers and English Language.
- The student may be required to scan the documents and send.

(III)USECASE DIAGRAM:

The Exam Registration use cases in our system are:
1. Login
2. View exam details
3. Register
4. Acknowledgement
5. Fee Processing

ACTORS INVOLVED:
1. Student
2. System DB

USE-CASE NAME: LOGIN

The student enters his username and password to login and retrieve the information'.

USE-CASE NAME: VIEW EXAM DETAILS

The student view the details about the exam schedule which contains Date,time,etc...

USE-CASE NAME: REGISTER

The student should notify the fee details that only the student can pay the correct amount

USE-CASE NAME: ACKNOWLEDGEMENT

The exam fees should be paid by the student to get the hall ticket from the exam controller.

USE-CASE NAME: FEE PROCESSING

All the details should be viewed by both the student and the controller to verify whether all the entered details are correct.
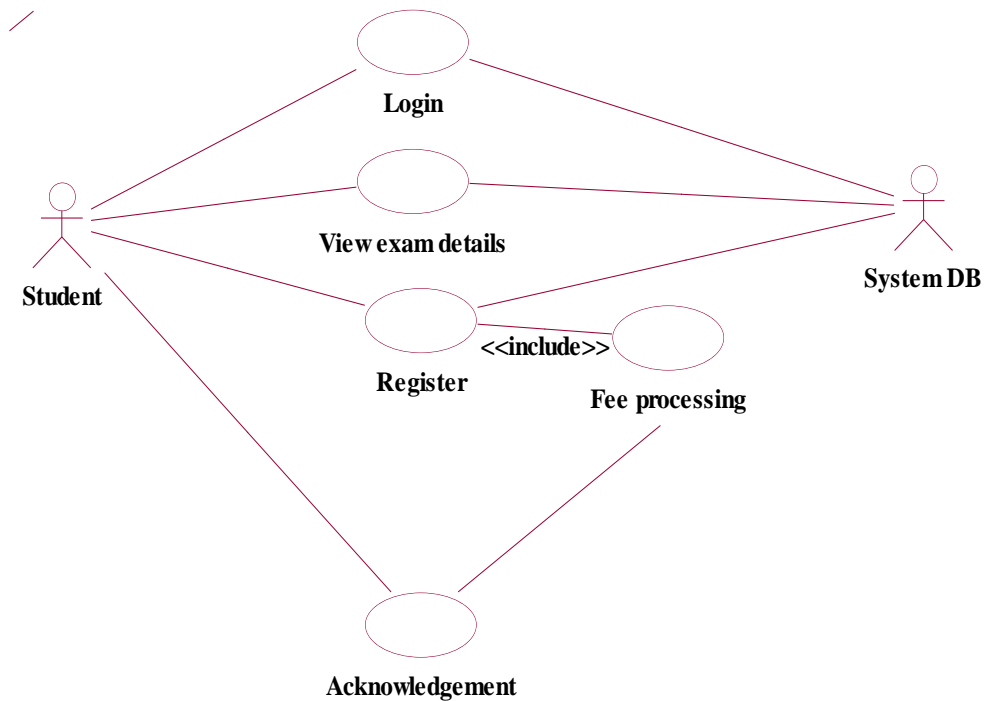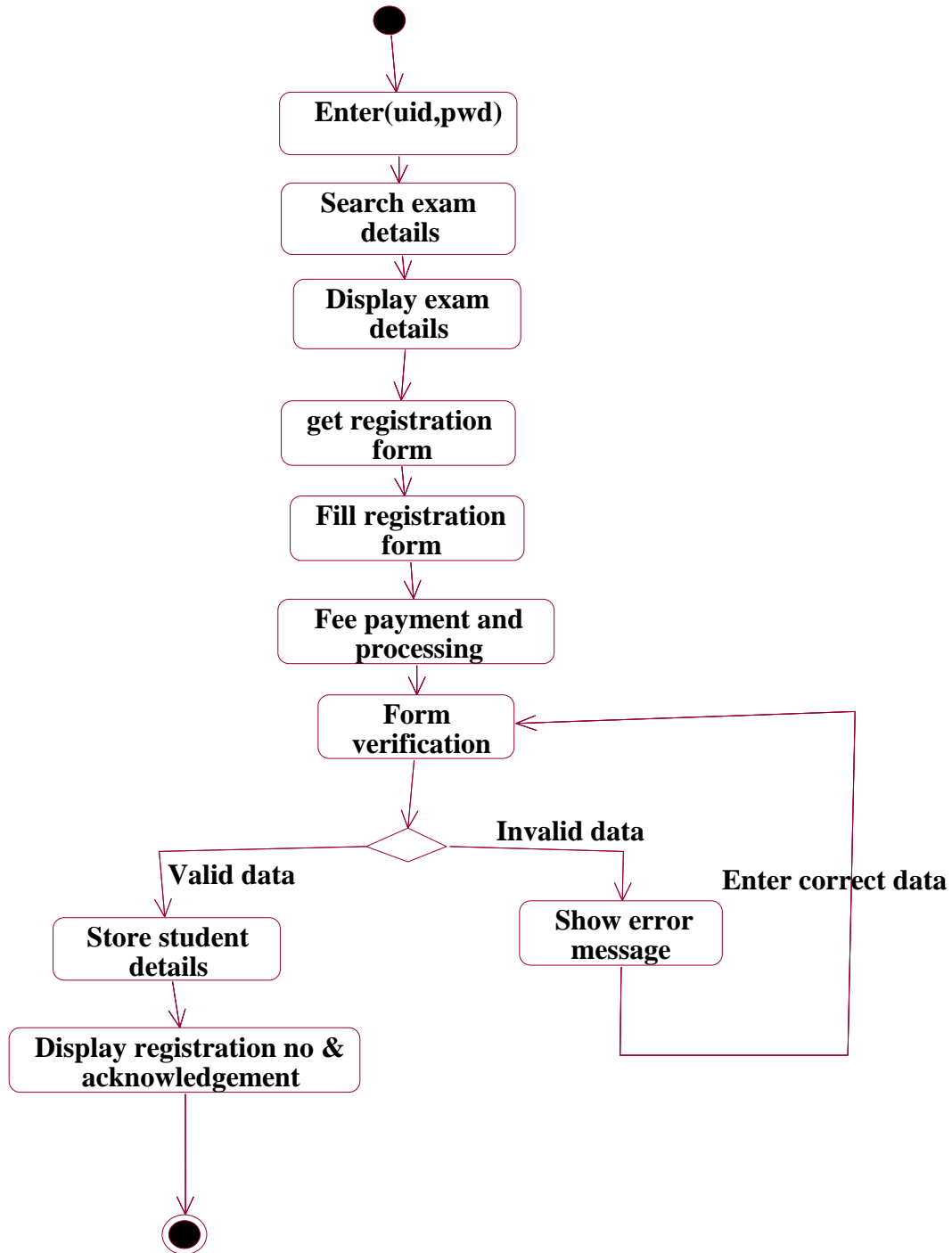
Fig. 3. 1. USECASE DIAGRAM FOR EXAM REGISTRATION SYSTEM

(IV) ACTIVITY DIAGRAM:

Enter(uid,pwd)

Search exam details

Display exam details

get registration form

Fill registration form

Fee payment and processing

Form verification

Valid data

Invalid data

Enter correct data

Store student details

Show error message

Display registration no & acknowledgement

(V)CLASS DIAGRAM:

The class diagram, also referred to as object modeling is the main static analysis diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects.

The Exam Registration System class diagram consists of four two classes of registration system.

1. Student_details
2. Exam_details
3. Register

1) STUDENT_DETAILS

It consists of six attributes and six operations. The attributes id, password, name, age, sex, course. The operations of this class are login(), logout(), conformation(), register(), newfeesdetails().

2) EXAM_DETAILS

It consists of four attributes and six methods. The attributes are userid, password, examfees, fees due. The methods are login(),logout(), feesdetails(), displayfees(), conformation(), examcontroller().

3) REGISTER
This class is used to maintain the registered student information such as, subject registered, date of registration and etc,.
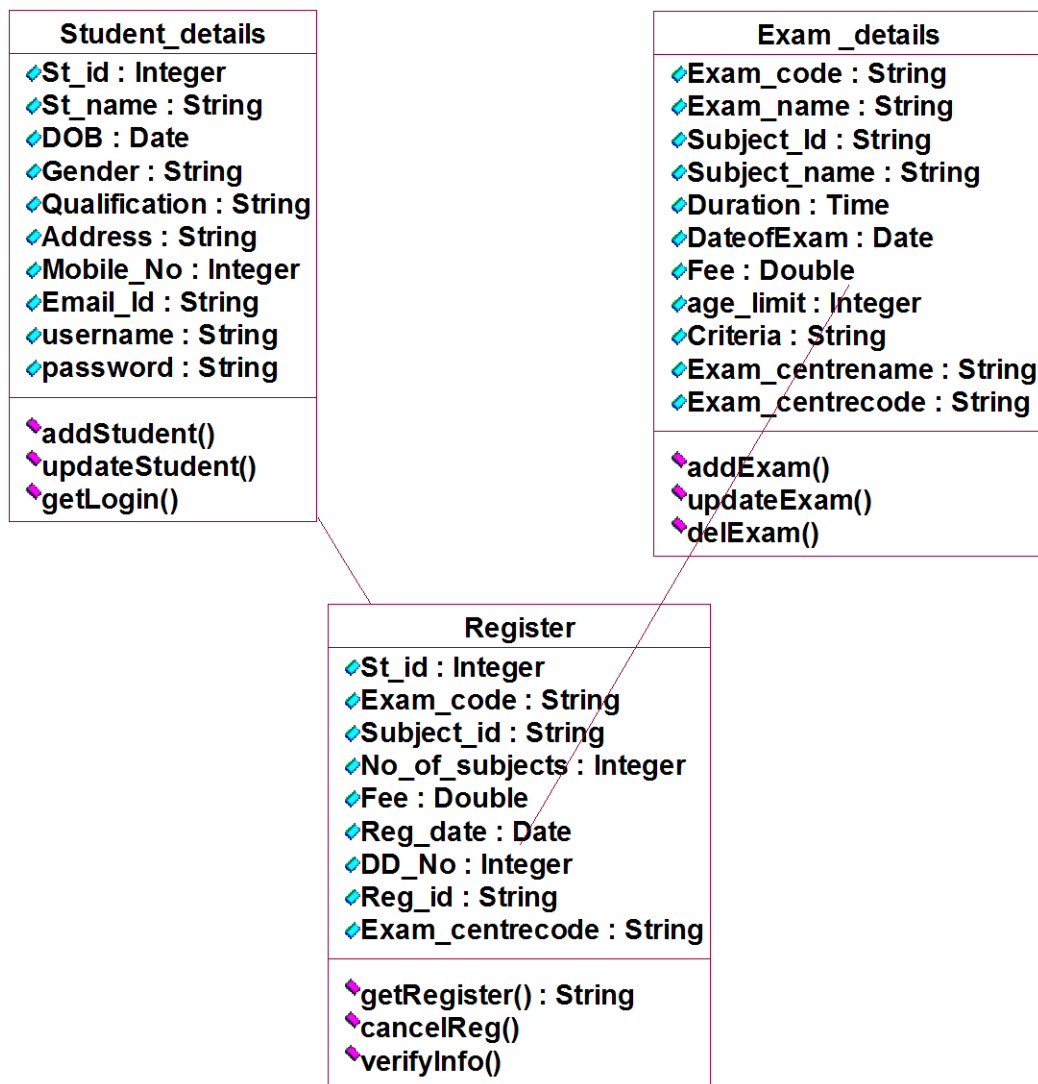
Student_details

◆St_id : Integer
◆St_name : String
◆DOB : Date
◆Gender : String
◆Qualification : String
◆Address : String
◆Mobile_No : Integer
◆Email_Id : String
◆username : String
◆password : String

◆addStudent()
◆updateStudent()
◆getLogin()

Exam _details

◆Exam_code : String
◆Exam_name : String
◆Subject_Id : String
◆Subject_name : String
◆Duration : Time
◆DateofExam : Date
◆Fee : Double
◆age_limit : Integer
◆Criteria : String
◆Exam_centrename : String
◆Exam_centrecode : String

◆addExam()
◆updateExam()
◆delExam()

Register

◆St_id : Integer
◆Exam_code : String
◆Subject_id : String
◆No_of_subjects : Integer
◆Fee : Double
◆Reg_date : Date
◆DD_No : Integer
◆Reg_id : String
◆Exam_centrecode : String

◆getRegister() : String
◆cancelReg()
◆verifyInfo()

Fig.5.1 CLASS DIAGRAM FOR EXAM REGISTRATION SYSTEM

(VI)INTERACTION DIAGRAM:

A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices.

An event also is considered to be any action by an object that sends information. The event line represents a message sent from one object to another, in which the "form" object is requesting an operation be performed by the "to" object. The "to" object performs the operation using a method that the class contains.

It is also represented by the order in which things occur and how the objects in the system send message to one another.
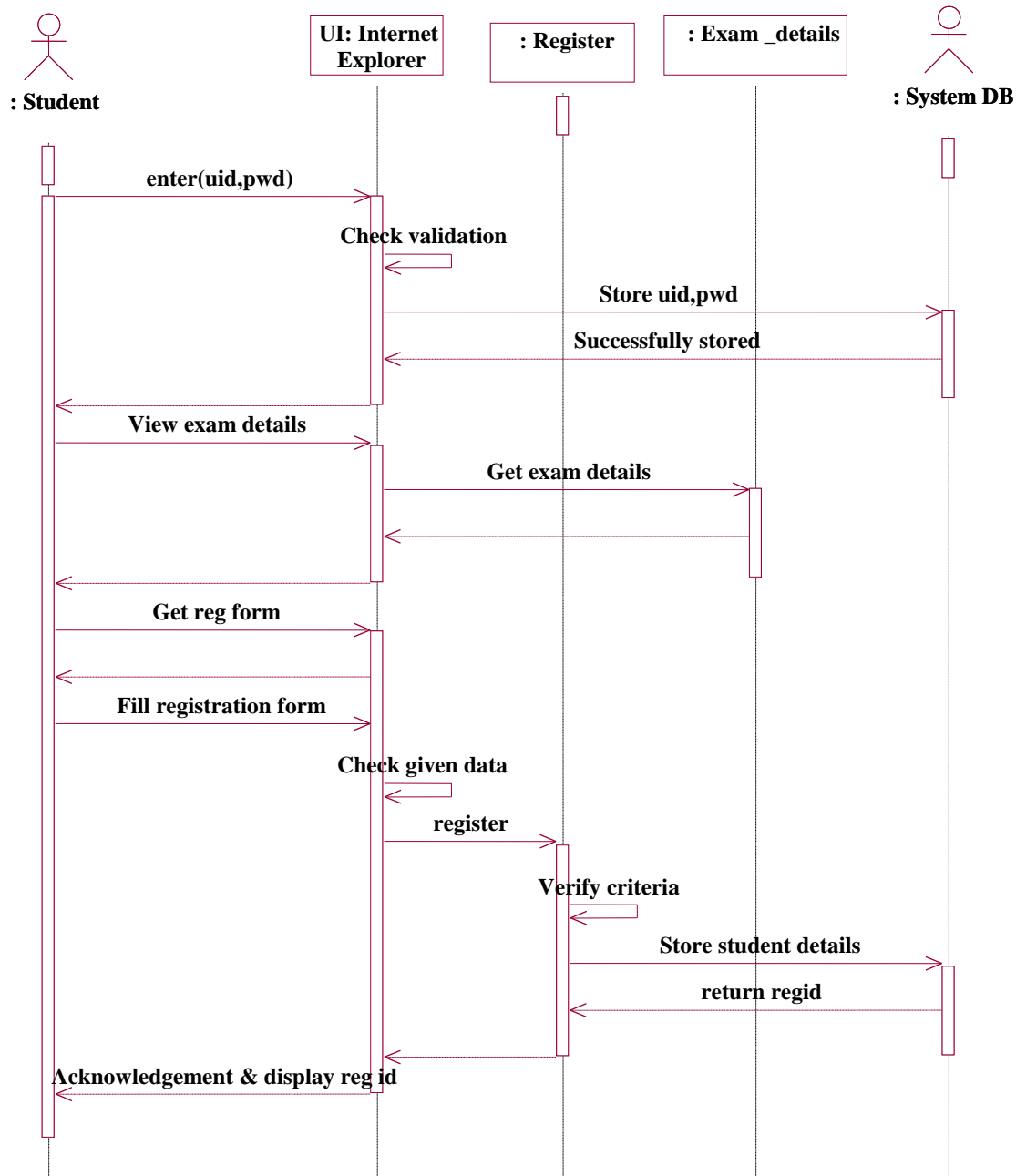
Fig. 6.1. SEQUENCE DIAGRAM FOR REGISTRATION SYSTEM

The sequence and collaboration diagram represents that the student enter the information to get the hall ticket and the exam controller issues the hall ticket after verifying the necessary items and this data are stored in the database.
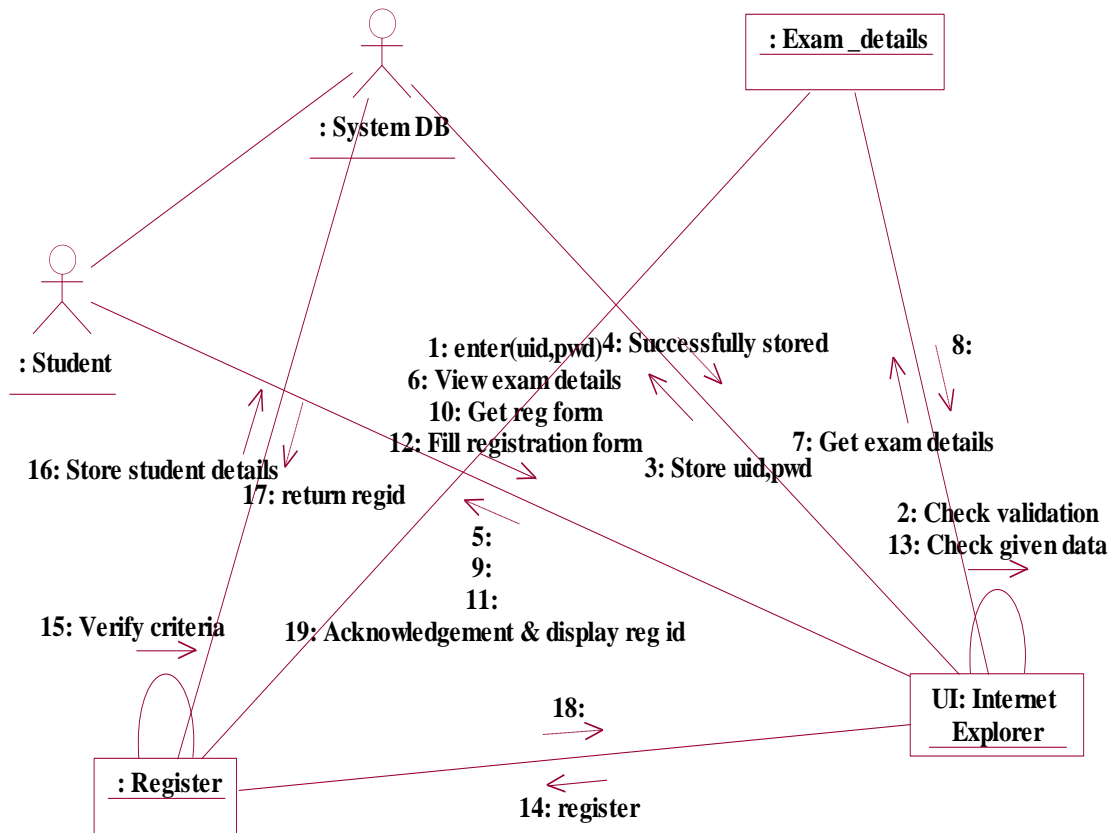
: Exam _details

: System DB

: Student

1: enter(uid,pwd) 4: Successfully stored

6: View exam details

10: Get reg form

12: Fill registration form

8:

7: Get exam details

3: Store uid,pwd

16: Store student details

17: return regid

5:

9:

11:

19: Acknowledgement & display reg id

2: Check validation

13: Check given data

15: Verify criteria

18:

UI: Internet
Explorer

: Register

14: register

Fig. 6.2. COLLABORATION DIAGRAM FOR REGISTRATION SYSTEM

(VII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.
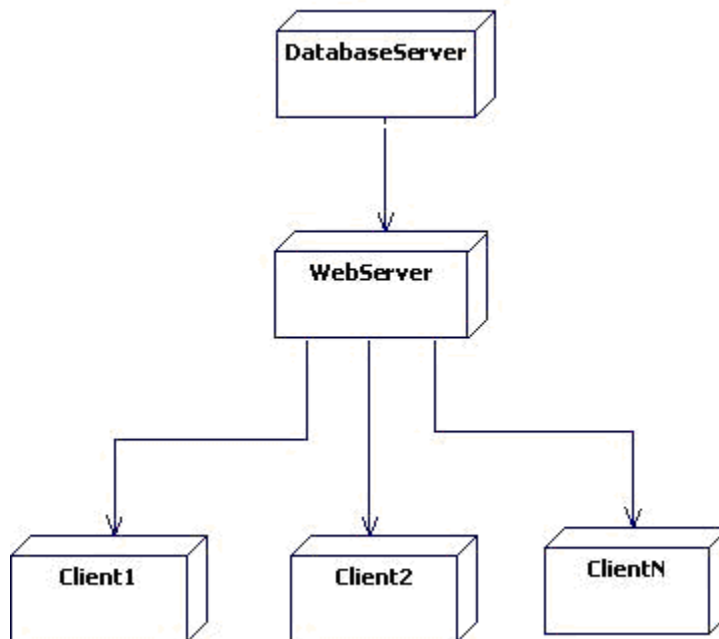
Fig.7.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationships among components in a system.
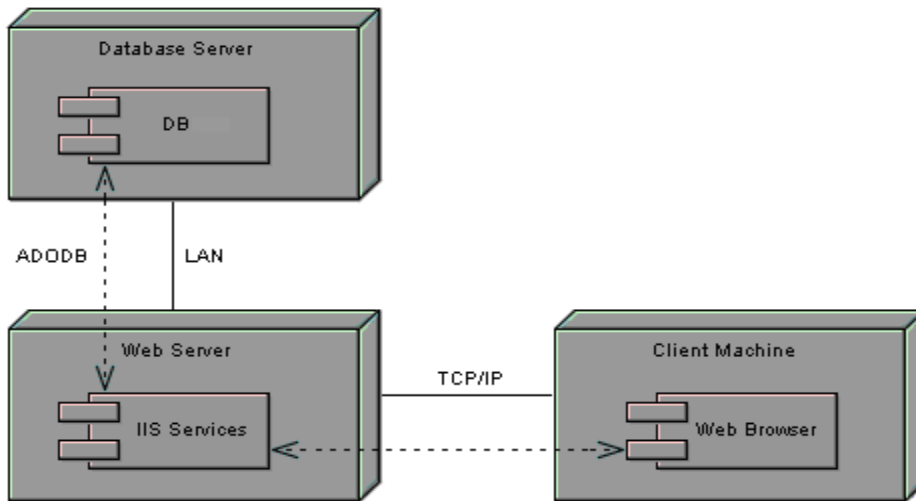
Fig.7.2.COMPONENT DIAGRAM

(VIII) IMPLEMENTATION OF DOMAIN OBJECTS LAYER AND TECHNICAL SERVICE LAYER
//Source file: D:\\uma\\examDetails.java

public class examDetails
{
  private string examCode;
  private string examName;
  private string subjectId;
  private string subjectName;
  private time duration;
  private date dateOfExam;
  private double fee;
  private integer ageLimit;
  private string criteria;
  private string examCentreName;
  private string examCentreCode;
  public studentDetails theStudentDetails;
  public register theRegister;

  /**
   * @roseuid 515AA57101B5
   */
  public examDetails()
  {

  }

```
/**
 * @roseuid 515AA448037A
 */
public void addExam()
{

}

/**
 * @roseuid 515AA44F00BB
 */
public void updateExam()
{

}

/**
 * @roseuid 515AA4570280
 */
public void delExam()
{

}
}
//void examDetails.deleteExam(){
//
//   }
```

//*Source file: F:\\Vaish\\Register.java*

```
public class Register
{
  private int studid;
  private String ExamCode;
  private String subid;
  private int no.ofSubject;
  private double fees;
  private string regid;
  private String ExamCenterCode;

  /**
  @roseuid 51342F76033C
   */
  public Register()
```

```
    {

    }

    /**
    @roseuid 51342C88004E
     */
    public void getRegister()
    {

    }

    /**
    @roseuid 51342C8E0271
     */
    public void cancelRegister()
    {

    }

    /**
    @roseuid 51342CA20109
     */
    public void verifyIngormation()
    {

    }
}
//register.register()
//register.getregister(){
//    return null;
//   }
//register.cancelreg(){
//
//   }
//register.verifyinfo(){
//
//   }


//Source file: F:\\Vaish\\StudentDetails.java

public class StudentDetails
{
    private string Studname;
    private integer Studid;
```

```
private Date DOB;
private String gender;
private String qualification;
private string Address;
private integer mobileno;
private string emailid;
private string username;
private string password;

/**
@roseuid 51342F7602CE
 */
public StudentDetails()
{

}

/**
@roseuid 51342B4901E4
 */
public void addStudent()
{

}

/**
@roseuid 51342B4F03A9
 */
public void updateStudent()
{

}

/**
@roseuid 51342B58029F
 */
public void getLogic()
{

}
}
/**
void studentdetails.getlogin(){

  }
studentdetails.studentdetails()
```
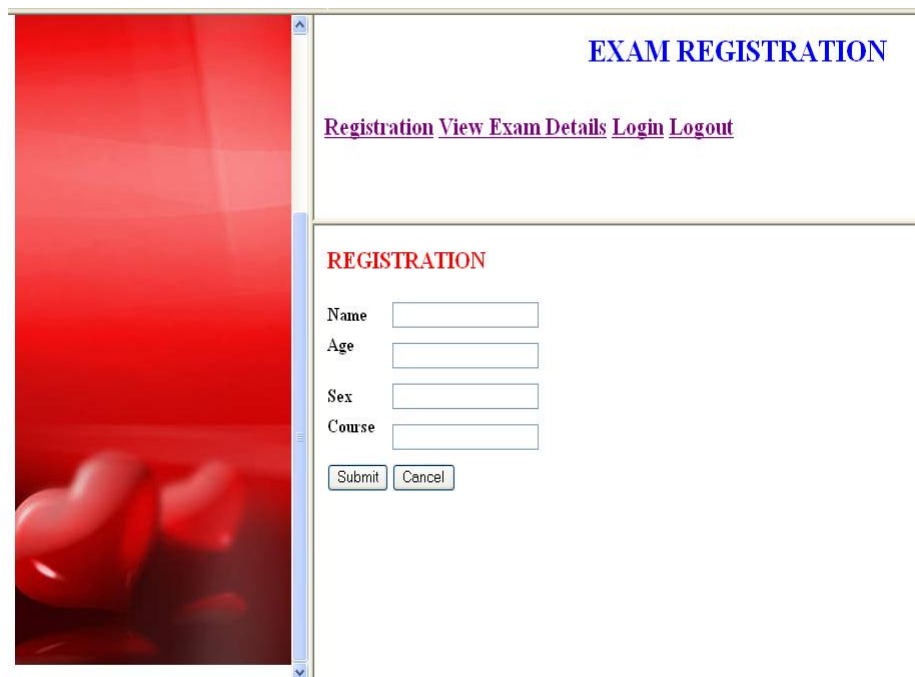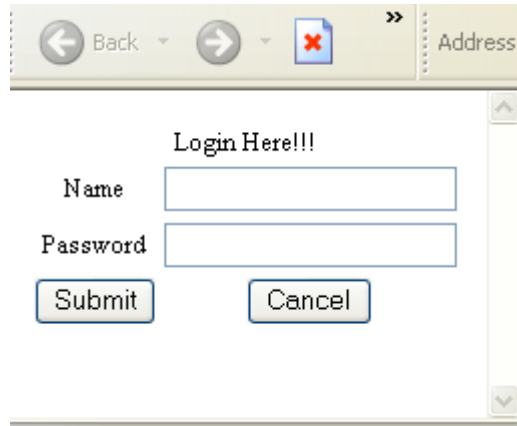
void studentdetails.updatestudent( ){

   }
void studentdetails.addstudent( ){

   }

 /

(IX) IMPLEMENTATION OF USER INTERFACE LAYER

Fig.9.1. REGISTRATION FORM

Fig.9.2 LOGIN FORM

RESULT:

Thus the mini project for Exam Registration system has been successfully executed and codes are generated.