Virtual Classroom System

A Project Report
Submitted in the partial fulfillment for the award of the degree
Of

(B.Tech(CS) 3rd year)

Submitted by

Saloni Gupta(1812849)
Shikha Gupta(1812861)
Shivani Sharma(1812866)
Shreya Sinha(1812871)

Under the supervision of
Dr. Monika Saxena
(Department of Computer Science)

Department of computer Science
Banasthali Vidyapith
Banasthali - 304022
Session: 2020-21

Abstract

Virtual class room is an e-learning project that enables the traditional teaching process to be done using a website through internet. The virtual class room helps students to attend their lectures from anywhere in the world, and provides more efficient technologies for instructors to communicate with the students and provide information that can be easily understood by the students. Our project is a website based on java and JSP, which provides a virtual class room for registered student and instructors, which can interact with each other.

Acknowledgement

We would like to express my special thanks to our mentor “Dr. Monika Saxena” for their able guidance support in completing our Project and for extending their valuable times for us to take up this problemas a project. on
“Virtual Classroom System”
Last but not the least I would like my gratitude to
Prof.C K Jha (H.O.D, Dept of BTech) and
Dr Neelam Sharma(Project Coordinator) and Dr. Deepak Kumar(Project Coordinator) who helped me in his way whenever needed and providing me with all the facility that was required.

Date: 12-05-2021

SaloniGupta(1812849)
Shikha Gupta(1812861)
Shivani Sharma(1812866)
Shreya Sinha(1812871)

Table of Contents

Objective

A virtual classroom aims to provide a learning experience that is similar to a real classroom. Virtual classroom enables to bring learners from around the world together in a highly interactive virtual class, greatly reducing the travel, time, expense of onsite programs. In the current situation due to COVID 19 it has emerged with a lot of possibilities. This project has a lot of scope for future development. Developing a virtual classroom system to promote a greater count of students to splurge into the field of Education.Virtual Classroom would definitely provide a new revolution in our education system scenario. The portability it provides is beneficial from a student’s point of view. The feature it has makes it a better educational tool for students. The methods that will be implicated are useful for learning and saving time and effort. The application provides real-time streaming of video and is portable so the range within which one can operate is reasonable. The Student can have the facilities of getting the other resources like videos and PPTs required for the academic education. The use of this application will embark a true mark in the education society and then learning would be affordable to distances.

Requirement Analysis (SRS)

- **Requirement specification**

  ### Hardware Requirements

  Processor: Dual Core or above 10
  Ram: 1GB (or) Higher
  Hard disk: 20GB

  ### Software Requirements

  Developer:
  Technology: Java SE<-, JDBC |Java EE<-Servlet, JSP
  Web Server: Tomcat 7.0
  Client Side Technologies: HTML, CSS, JavaScript
  Server Side Technologies: Servlets, JSP
  Data Base Server: DERBY
  Operating System: Microsoft Windows, Linux or Mac any version

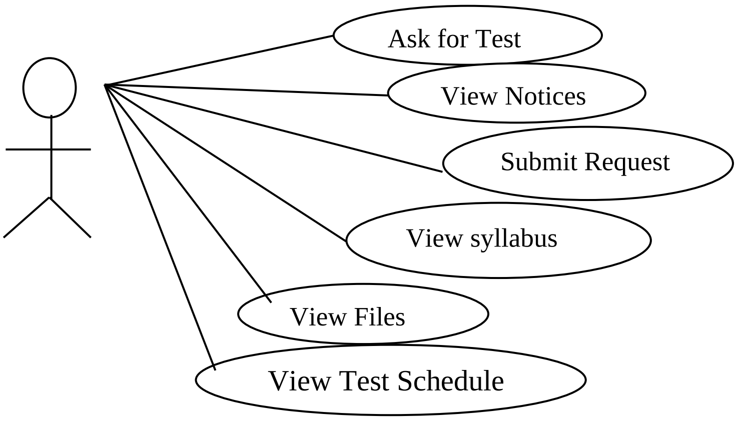
  User:
  Technology: Java SE<-, JDBC |Java EE<-Servlet, JSP
  Server Side Technologies: Servlets, JSP
  Data Base Server: DERBY
  Operating System: Microsoft Windows, Linux or Mac any version

- **Product Functions**
  - The instant messaging feature will allow users to interact with faculty and fellow classmates with text-based format.

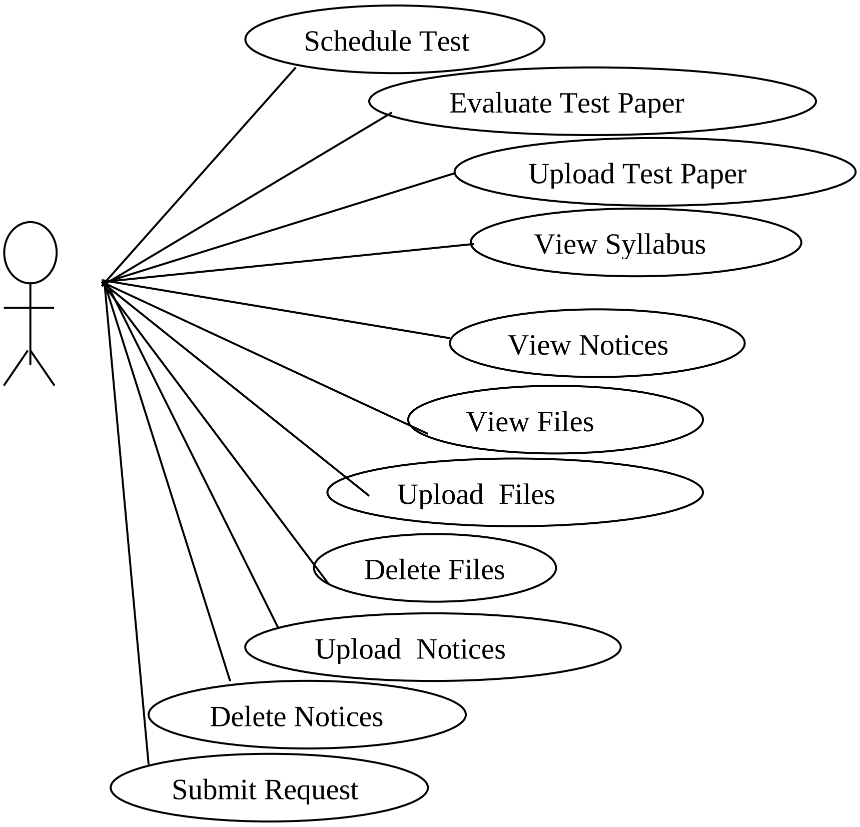
- File sharing and document collaboration tools allow users to efficiently share and develop documents and reports within a group.
- User profiles containing contact information, user id, and other related information on a student personal page. This page will only be accessible to students and faculty enrolled in the same class. Only the owner of the profile may changes to it.
- Users will have the ability to manage a private temporary hosting space Tools such as file uploading, directory creator, move, delete and rename will be accessible.

Use-case Diagrams

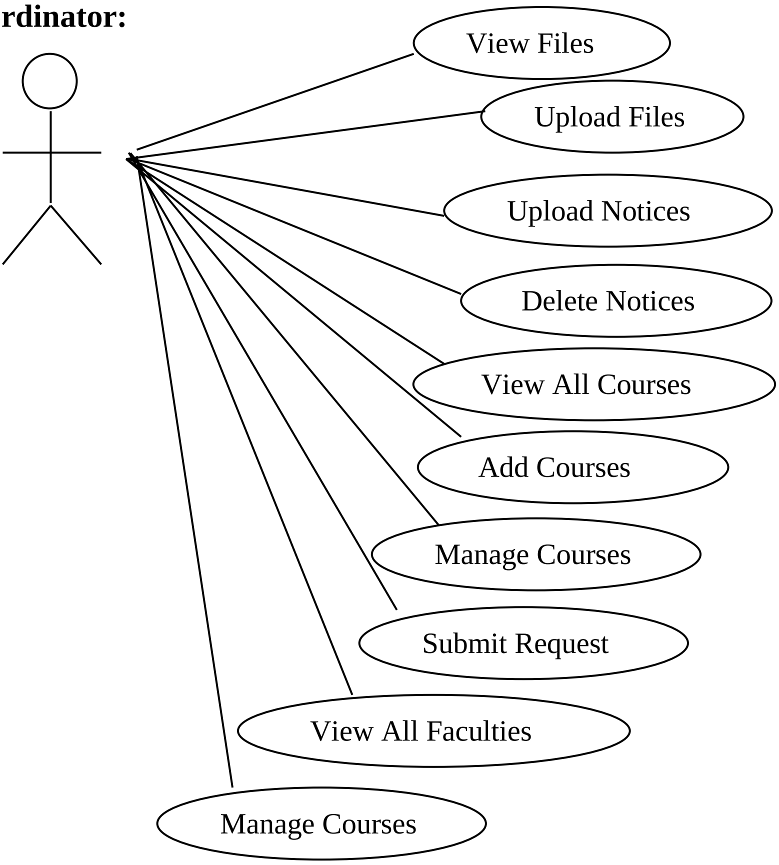
- **Student:**



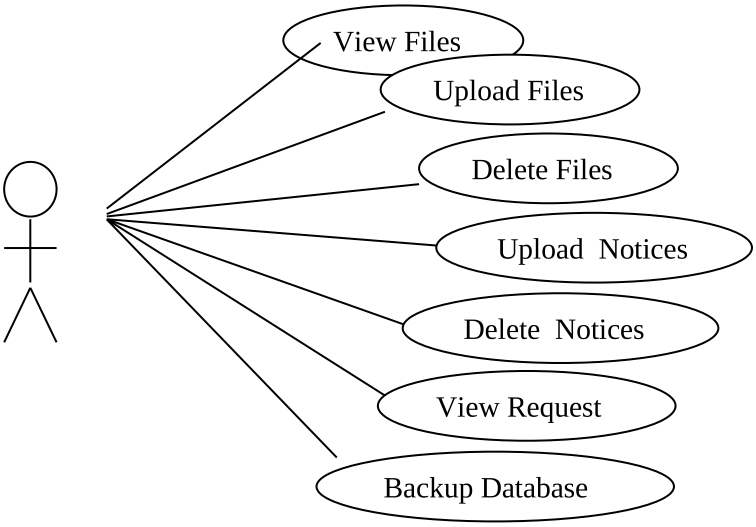
- **Faculty:**



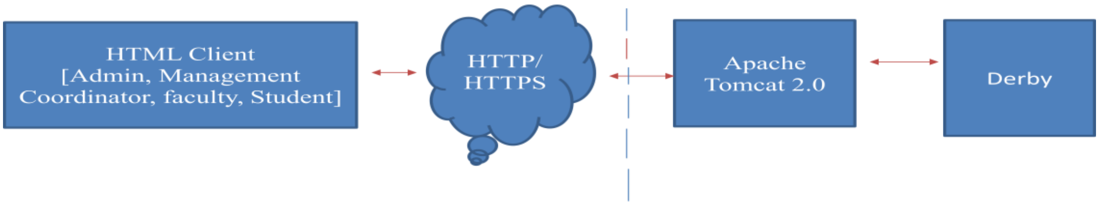
- **Management Coordinator:**



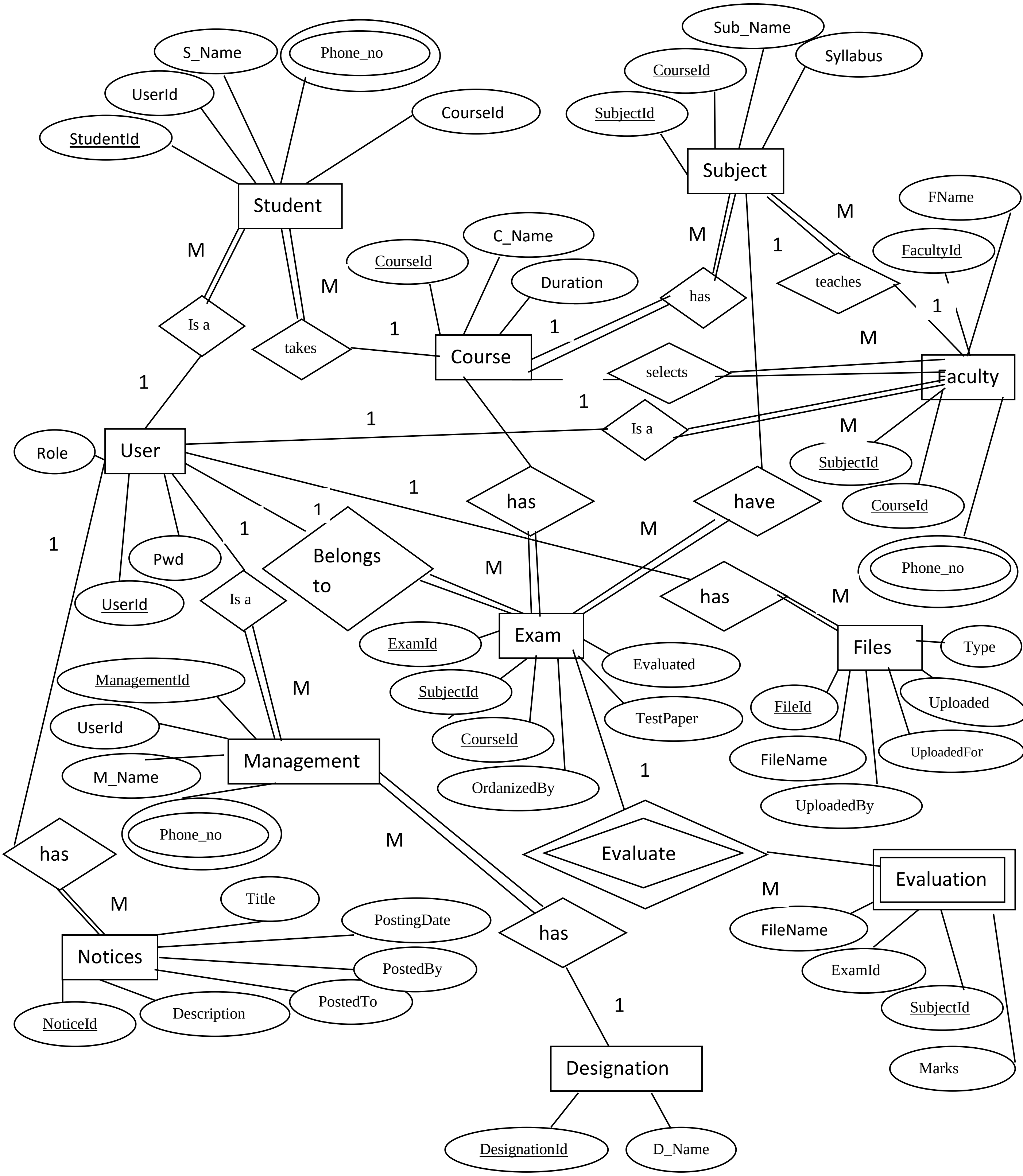
- **System Administrator:**



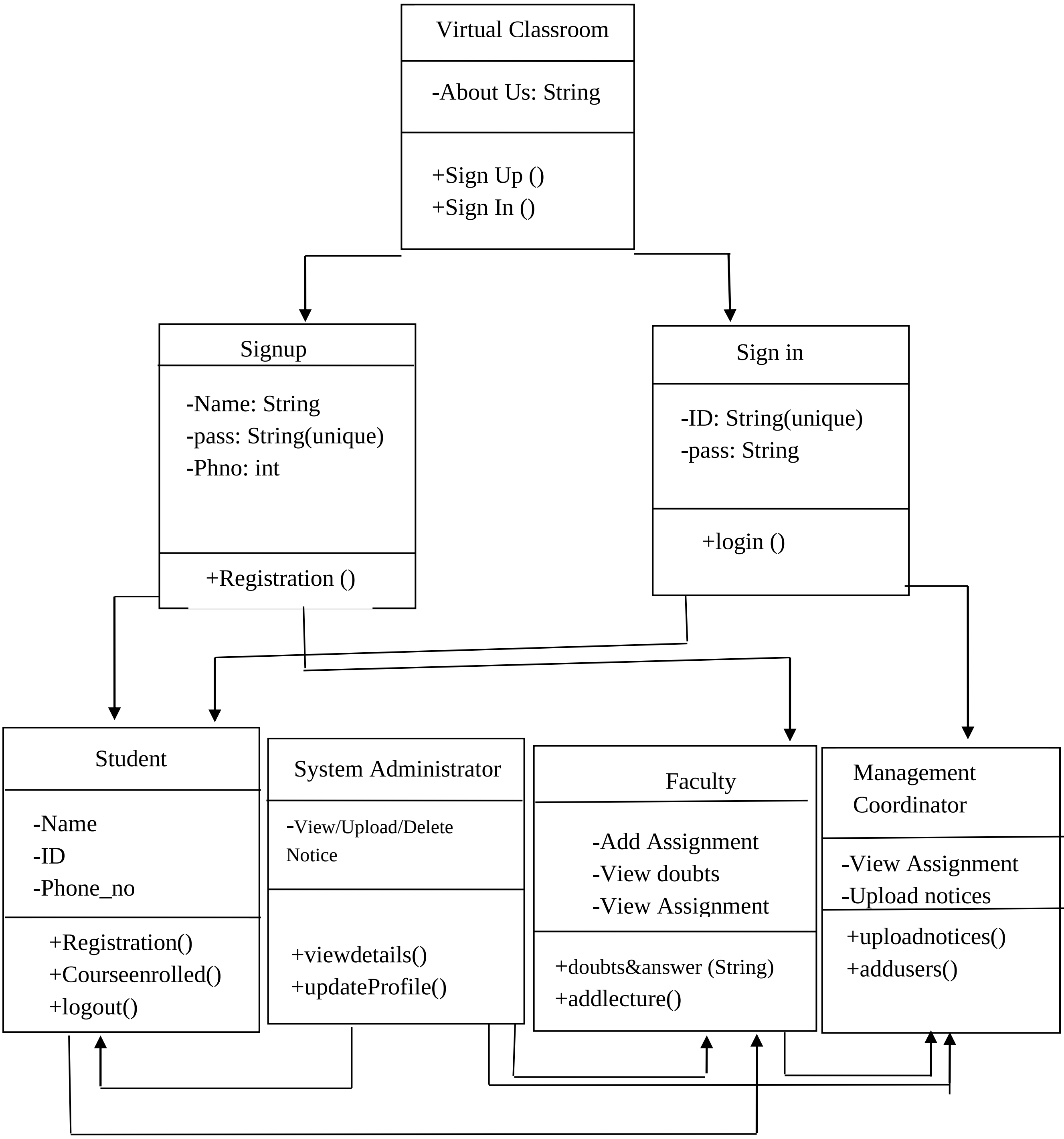
➢ System Design (SDS)



➢ ER Diagram


S_Name
Phone_no
UserId
StudentId
CourseId
Student
M
M
Is a
takes
1
1
Sub_Name
Syllabus
CourseId
SubjectId
Subject
M
M
1
has
teaches
FName
FacultyId
1
M
C_Name
CourseId
Duration
Course
1
selects
Faculty
1
Is a
M
SubjectId
CourseId
Phone_no
Role
User
1
1
1
1
1
Pwd
UserId
Is a
Belongs to
has
have
M
M
M
has
M
Files
Type
Uploaded
UploadedFor
FileId
FileName
UploadedBy
ExamId
Exam
Evaluated
SubjectId
TestPaper
CourseId
OrdanizedBy
1
ManagementId
UserId
M_Name
Management
M
Phone_no
M
Evaluate
Evaluation
M
FileName
ExamId
SubjectId
Marks
has
M
Title
PostingDate
Notices
PostedBy
PostedTo
Description
NoticeId
has
1
Designation
DesignationId
D_Name

- **Class Diagram**:



- **Database Design**

1. STUDENT

Field	Type	Null	Key	Default	Extra
StudentId	varchar(10)	No	Primary key	Not NULL	Student Id
UserId	varchar(10)	Yes	Foreign key	NULL	Valid user Id
S_Name	varchar(25)	Yes		NULL	Name of the student
phone_no	Integer(10)	Yes		NULL	Phone number of the student
CourseId	varchar(10)	Yes	Foreign Key		Course Id

2. FACULTY

Field	Type	Null	Key	Default	Extra
FacultyId	varchar(10)	No	Primary key	Not NULL	Faculty Id
UserId	varchar(10)	Yes	Foreign key	NULL	Valid User Id
F_Name	varchar(25)	Yes		NULL	Name of the faculty
phone_no	varchar(10)	Yes		NULL	Phone number of the faculty
CourseId	varchar(10)	Yes	Foreign Key	NULL	Course Id
SubjectId	varchar(10)	Yes	Foreign Key	NULL	Subject Id

3.Course

Field	Type	Null	Key	Default	Extra
CourseId	Varchar(10)	No	Primary Key	Not NULL	Course Id
C_Name	Varchar(25)	Yes		NULL	Name of the Course
Duration	Integer	Yes		NULL	Duration of the Course

4. Subjects

Field	Type	Null	Key	Default	Extra
SubjectId	Varchar(10)	No	Primary key	Not NULL	Subject Id
CourseId	Varchar(10)	Yes	Foreign key	NULL	Course Id
Sub_Name	Varchar(25)	Yes		NULL	Subject Name
Syllabus	Varchar(50)	Yes		NULL	Syllabus of the subject

5.Evaluation

Field	Type	Null	Key	Default	Extra
ExamId	Varchar(10)	Yes	Foreign key	Not NULL	Exam Id
StudentId	Varchar(10)	No	Foreign key	Not NULL	Student Id
SubmissionDate	Date	Null		NULL	Submission date
Filename	Varchar(10)	Null		NULL	Filename of assignment
Marks	Integer	Null		NULL	Marks of the Exam

6.Exam

Field	Type	Null	Key	Default	Extra
ExamId	Varchar(10)	No	Primary key	Not NULL	Exam Id
CourseId	Varchar(10)	Yes	Foreign key	NULL	Course Id
SubjectId	Varchar(10)	Not Null	Foreign key	NULL	Subject Id
OrganizedBy	Varchar(100)	Null		NULL	Who organized the exam
TestPaper_url	Varchar(100)	Null		NULL	Test paper of the exam
Marks	Integer	Null		NULL	Marks of the Subject

7. Notices

Field	Type	Null	Key	Default	Extra
NoticeId	Varchar(10)	No	Primary key	Not NULL	Notice Id
Description	Varchar(100)	Yes		NULL	Detailing of notice
PostedTo	Varchar(25)	Yes		NULL	To whom notices are posted
PostedBy	Varchar(25)	Yes		NULL	By whom notices are posted
PostingDate	Date	Yes		NULL	On what date notices have posted
Title	Varchar(25)	Yes		NULL	Title of the notice

8. Files

Field	Type	Null	Key	Default	Extra
FileId	Varchar(10)	No	Primary Key	Not NULL	File Id
Filename	Varchar(25)	Yes		NULL	Name of the file
UploadedBy	Varchar(25)	Yes		NULL	By whom files are uploaded
UploadedTo	Varchar(25)	Yes	Foreign key	NULL	To whom files are uploaded
UploadDate	Date	Yes		NULL	Date of uploading of files
Type	Varchar(10)	Yes		NULL	Which type of file it is

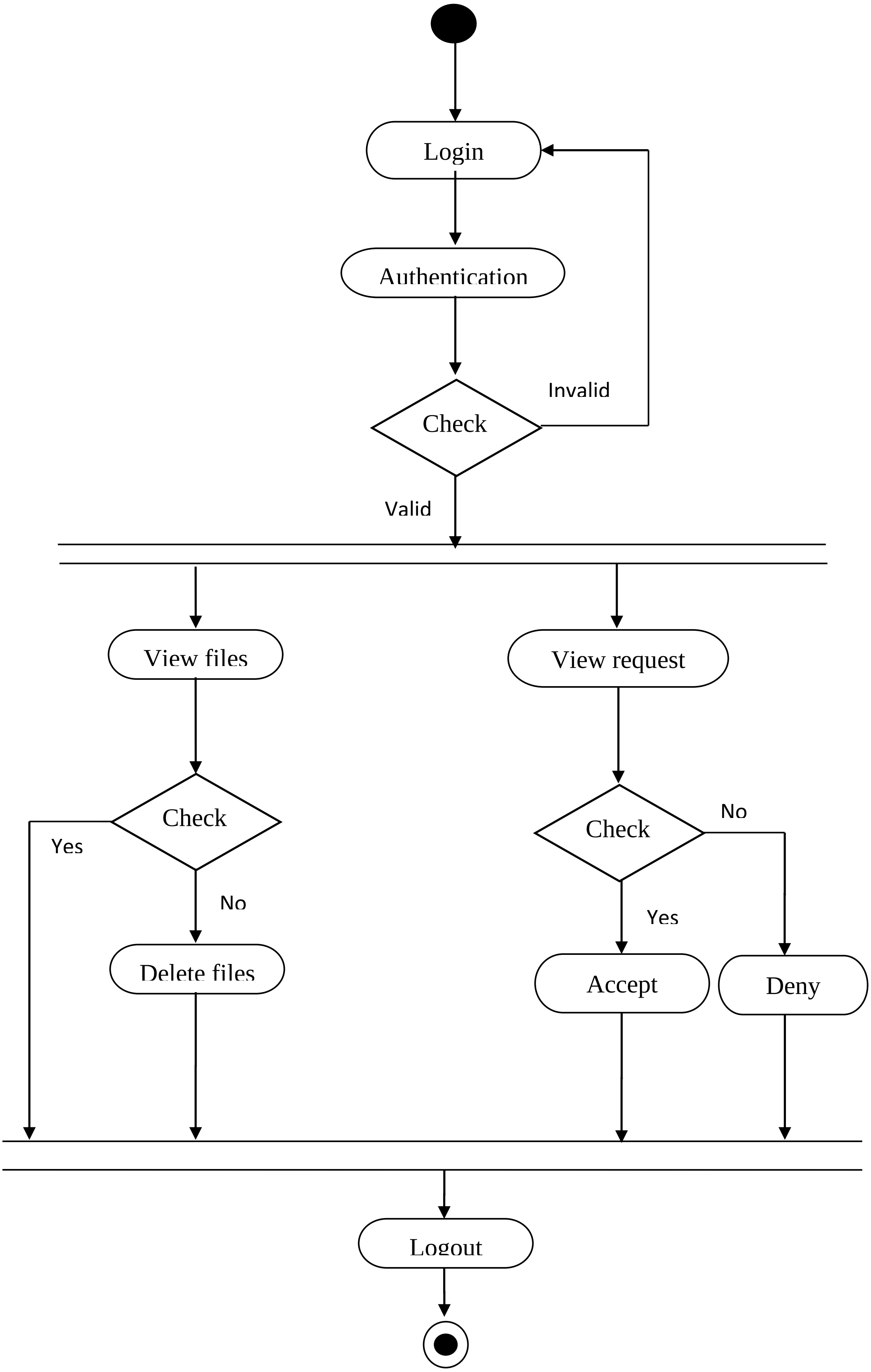
9. Management Coordinator

Field	Type	Null	Key	Default	Extra
Management Id	Varchar(10)	No	Primary key	Not NULL	Management Id
M_Name	Varchar(20)	Yes		NULL	Name of the management
Phone_no	Integer	Yes		NULL	Phone number of management
DesignationId	Varchar(15)	Yes	Foreign key	NULL	Designation id of the management
UserId	**Varchar(10)**	**Yes**	**Foreign key**	**NULL**	**User Id**

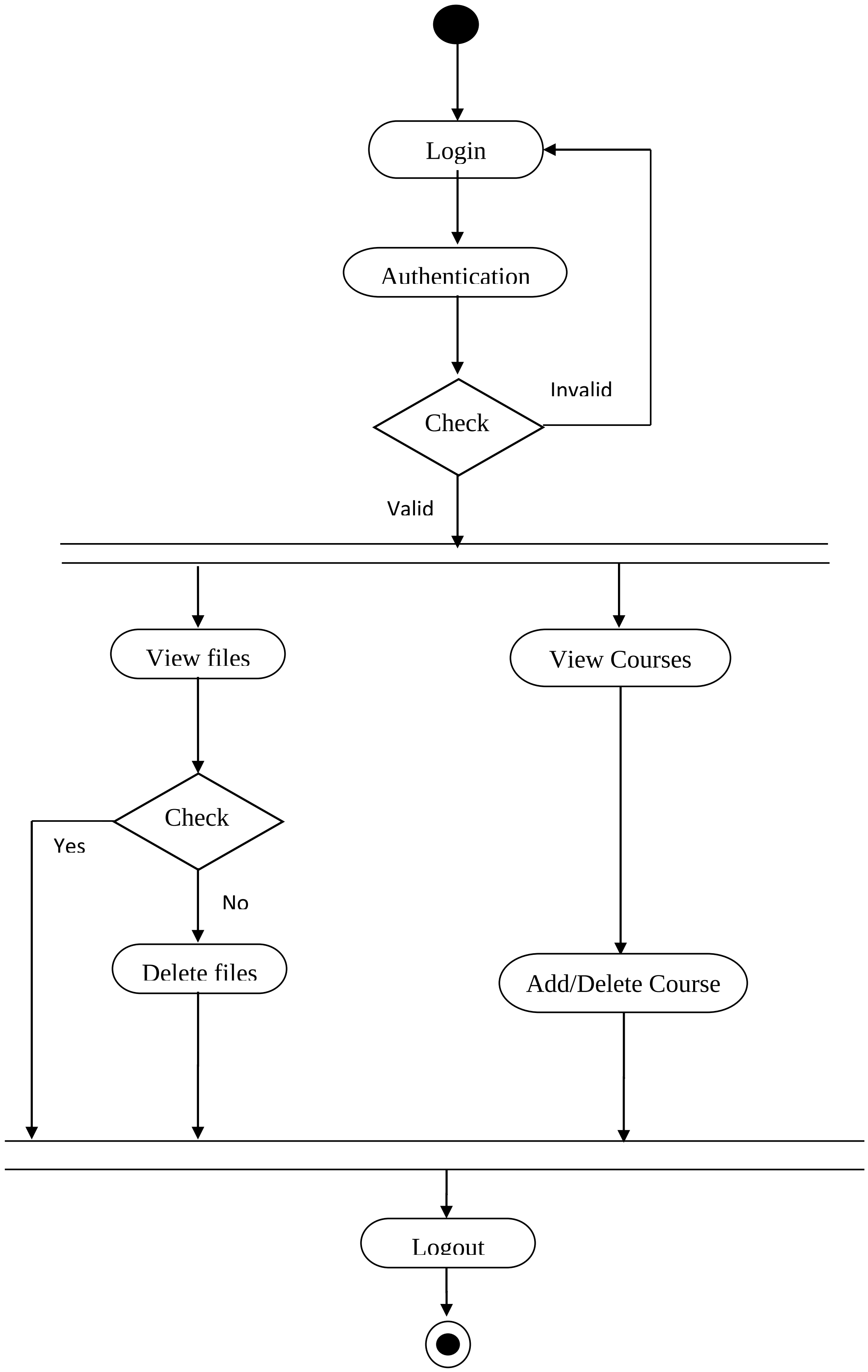
10. Users

Field	Type	Null	Key	Default	Extra
UserId	Varchar(10)	No	Primary key	Not NULL	User Id
Pwd	Varchar(10)	Yes		NULL	Password of the user
Role	Varchar(20)	Yes		NULL	Role of the user

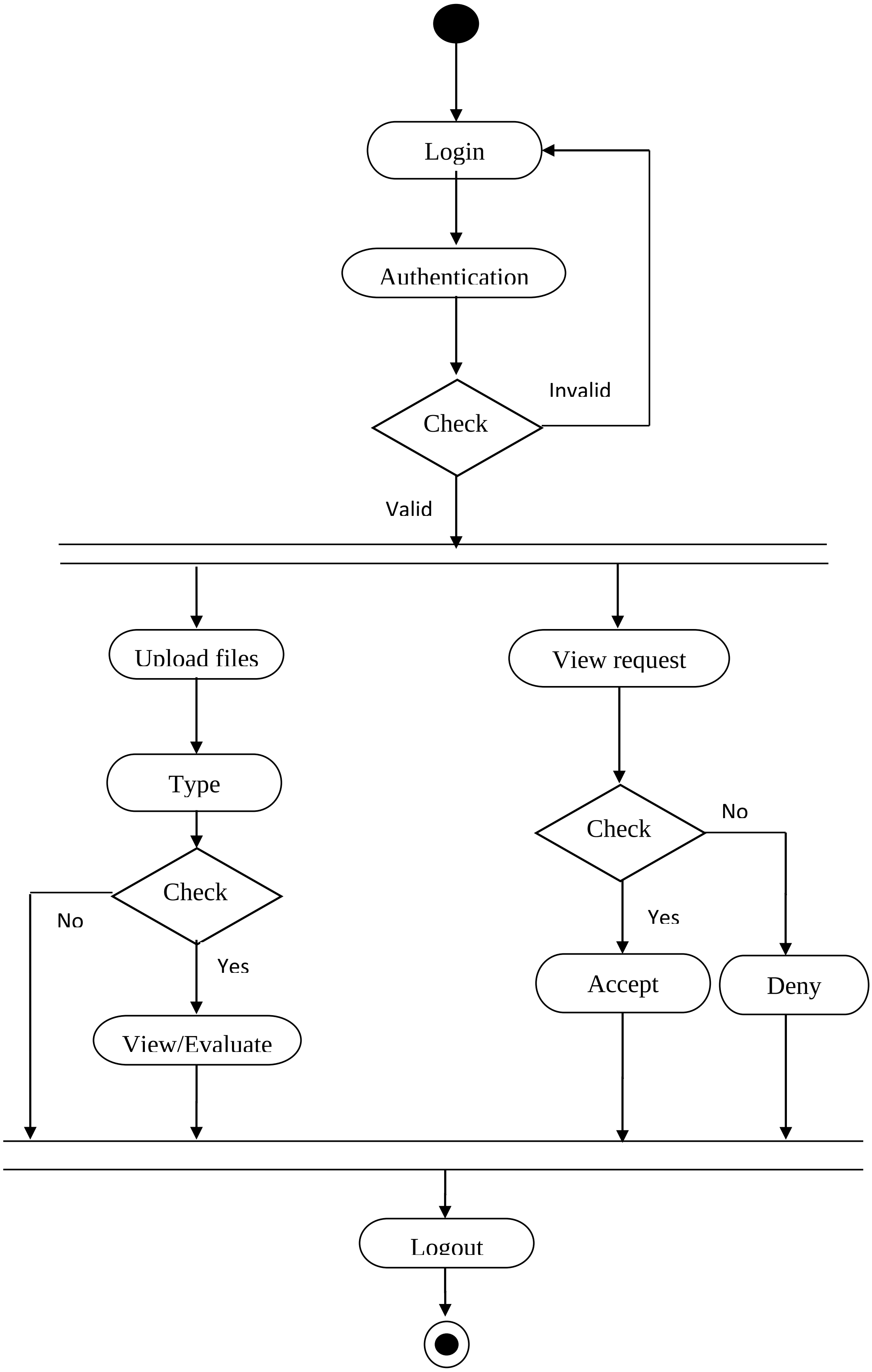
- **Activity Diagrams**
  - **SYSTEM ADMINISTRATOR**



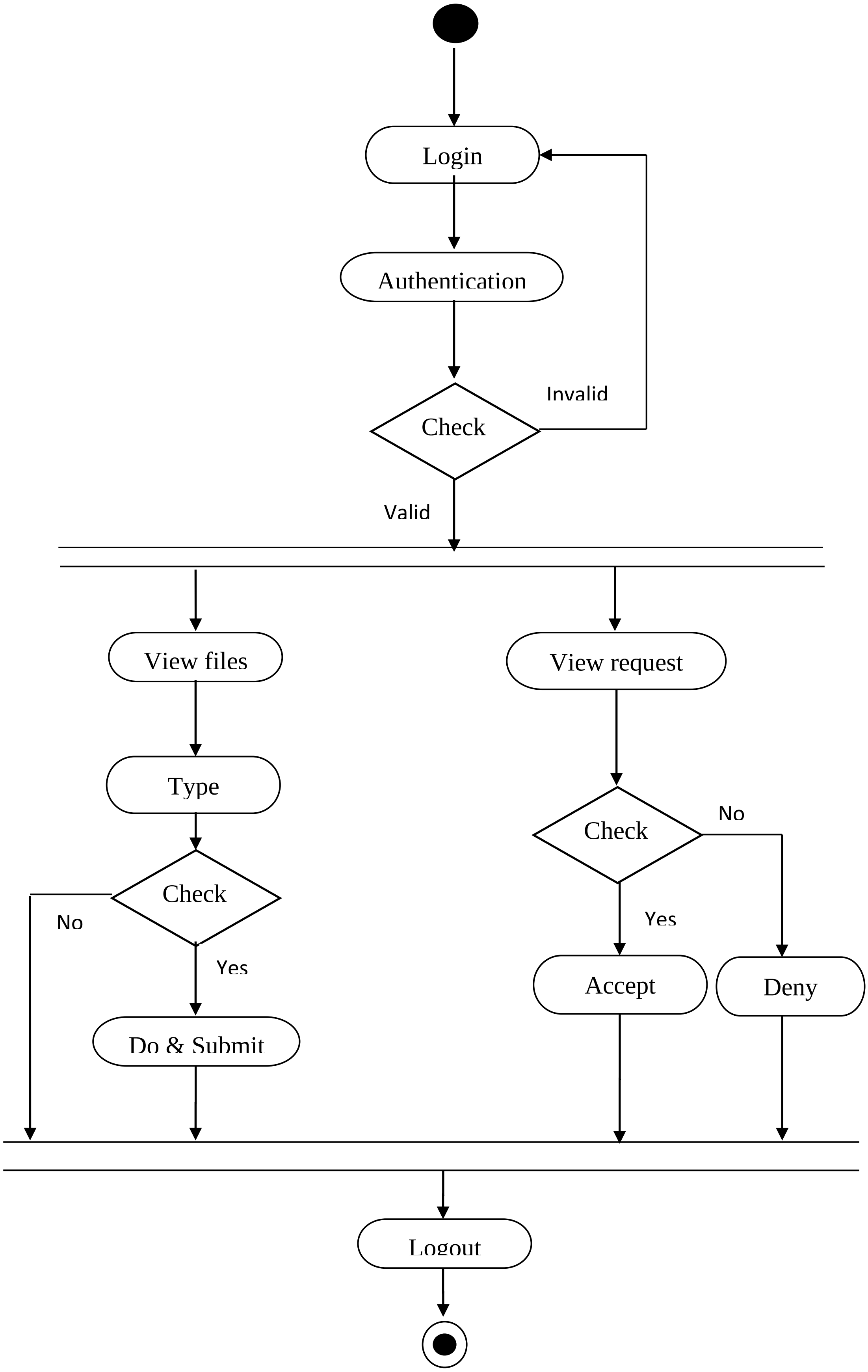
- **MANAGEMENT COORDINATOR**



- **FACULTY**

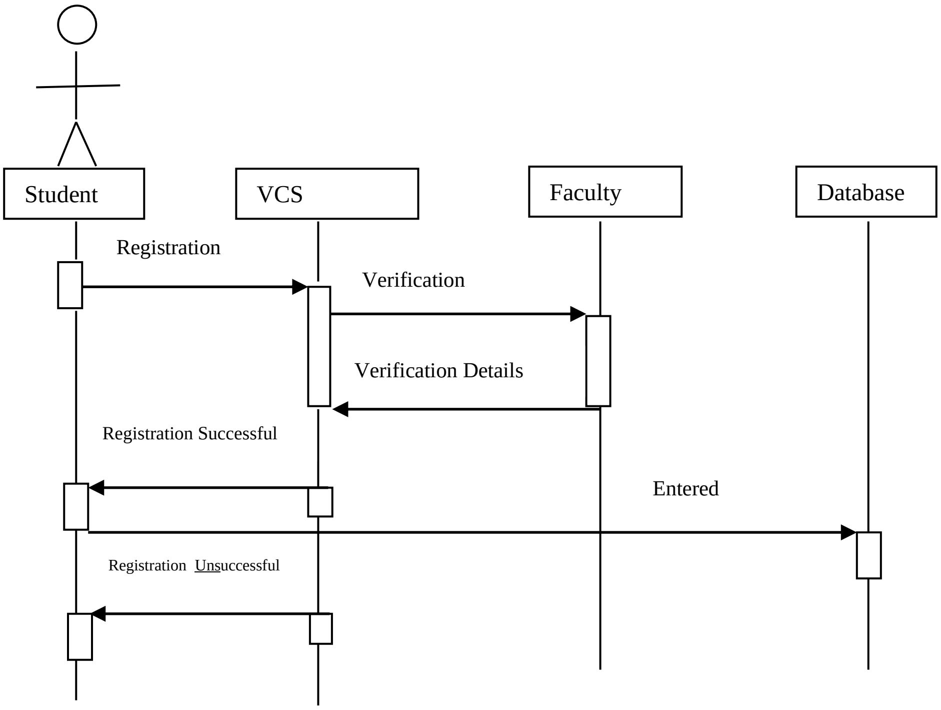

Login
Authentication
Check
Invalid
Valid
Upload files
View request
Type
Check
No
Yes
View/Evaluate
Check
No
Yes
Accept
Deny
Logout

- **STUDENT**

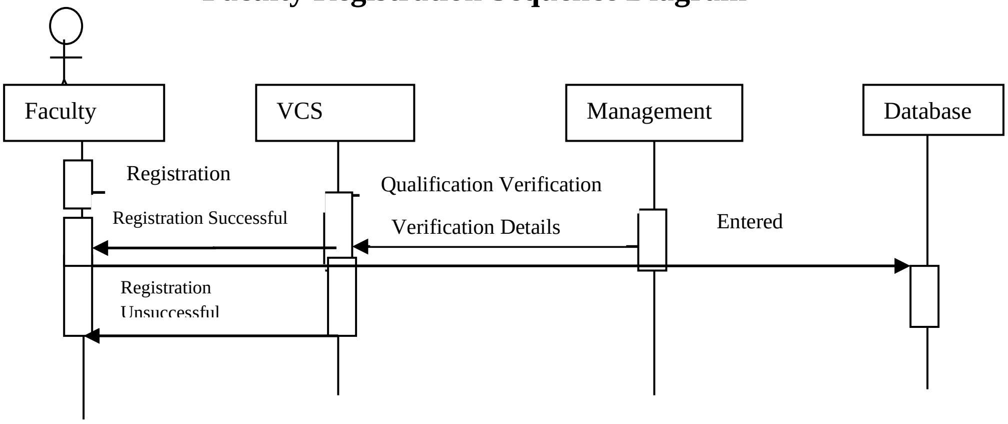

Login
Authentication
Check
Invalid
Valid
View files
View request
Type
Check
No
Yes
Do & Submit
Check
No
Yes
Accept
Deny
Logout

- **Sequence Diagrams**

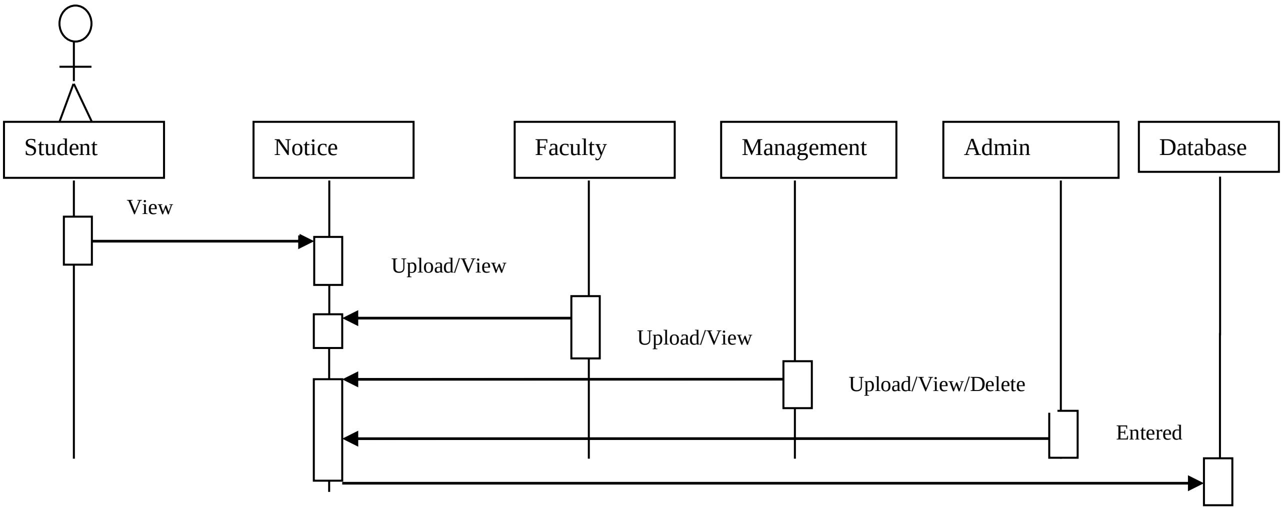
- **Student Registration Sequence diagram**



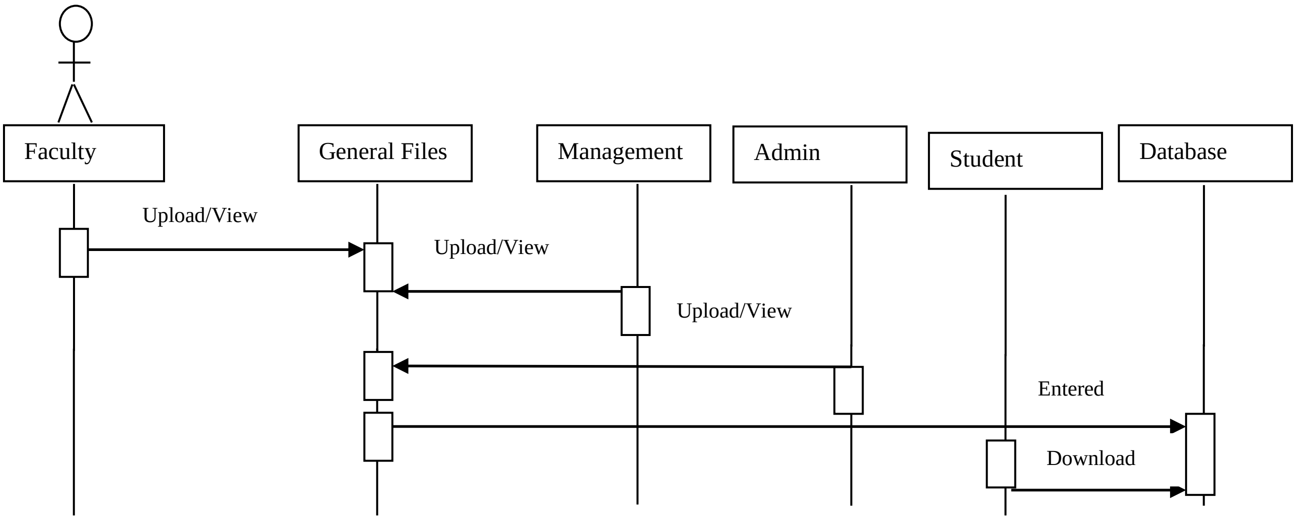
- **Faculty Registration Sequence Diagram**



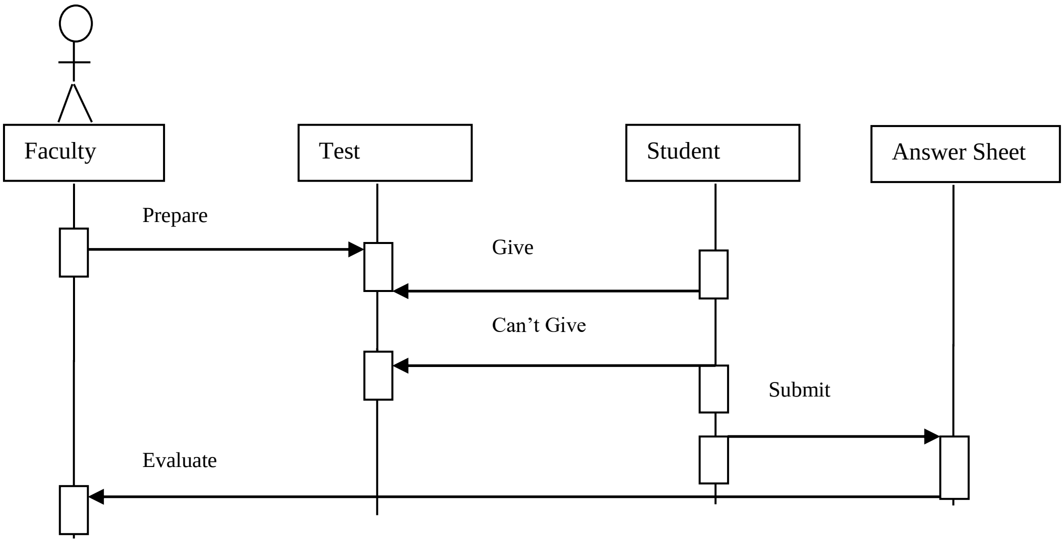
- **Notice Sequence Diagram**



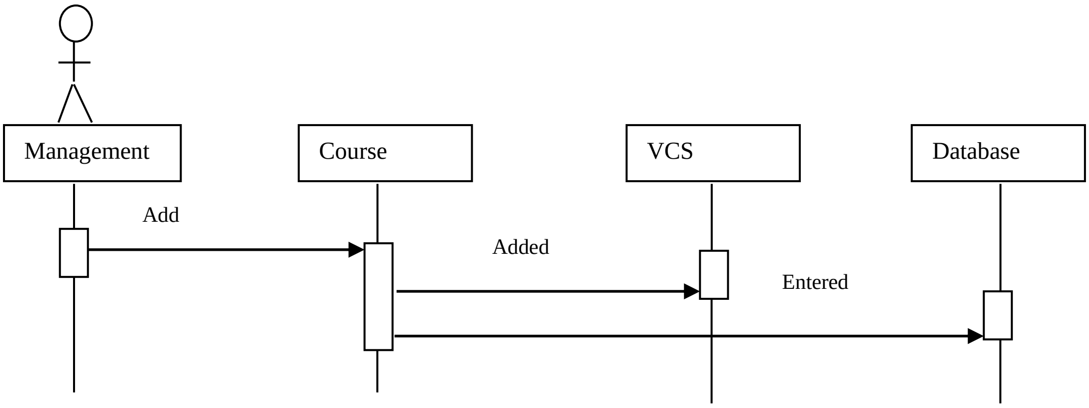
- **Files(General) Sequence Diagram**



- **Examination(Test) Sequence Diagram**



- **Adding Courses Sequence Diagram**



Coding

- **Home Page-Admin:**

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <center> <h3 style="background-color: pink;">ADMIN
HOME</h3></center>
    <nav>
        <ul>
<br></br><li
class="dropdown"><ahref="#">STUDENT<span>&rsaquo;</span>
</a>
 <ul>
             <li><a href="STUDENTVIEW_AD.jsp">VIEW</a></li>
 <li><a href="STUDENTREG_AD.jsp">ADD</a></li>
 <li><a href="delstud.jsp">DELETE</a></li>
 </ul>
      </li>
      <li class="dropdown"><a
href="#">FACULTY<span>&rsaquo;</span></a>
 <ul>
             <li><a href="FACULTYVIEW_AD.jsp">VIEW</a></li>
 <li><a href="FACULTYREG_AD.jsp">ADD</a></li>
 <li><a href="delfa.jsp">DELETE</a></li>
 </ul>
          </li>
        <li
class="dropdown"><ahref="#">MANAGEMENT<span>&rsaquo;<
/span></a>
 <ul>
         <li><a href="MANAGEVIEW_AD.jsp">VIEW</a></li>
 <li><a href="MANAGEREG_AD.jsp">ADD</a></li>
 <li><a href="delma.jsp">DELETE</a></li>
 </ul>
         </li>
 <li class="dropdown"><a
href="#">COURSE<span>&rsaquo;</span></a>
 <ul>
         <li><a href="COURSEVIEW_AD.jsp">VIEW</a></li>
 <li><a href="COURSEREG_AD.jsp">ADD</a></li>
 <li><a href="delcourse.jsp">DELETE</a></li>
 </ul>
 </li>
 <li class="dropdown"><a
href="#">NOTICE<span>&rsaquo;</span></a>
 <ul>
         <li><a href="VIEWNOTICE_AD.jsp">VIEW</a></li>
 <li><a href="UPLOADNOTICE_AD.jsp">ADD</a></li>
 <li><a href="delnotice.jsp">DELETE</a></li>
 </ul>
 </li>
      <li class="dropdown"><a
href="adminlogout.jsp">LOGOUT</a></li>
        </ul>
       </nav>
     </body>
</html>
```

- **Faculty Registration**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>new registration</title>
</head>
<body background="imgreg.jpg" >
   <center> <h3 style="background-color: pink;">FACULTY
REGISTRATION FORM</h3>
<form action="facreg.jsp" method="post">
   <table>
```

```
         <tr>
            <td>Faculty ID:<font style="color:red">*</font></td>
            <td><input type="text" name="use"  required  /></td>
         </tr>
         <tr>
         <td> Name :<font style="color:red">*</font></td>
         <td><input type="text" name="name"  required  /></td>
        </tr>
         <tr>
             <td>Password :<font style="color:red">*</font></td>
             <td><input type="password" name="pass" required
/></td>
         </tr>
        <tr>
            <td>Phone no :</td>
            <td><input type="text" name="ph_no" required   />
            <font style="color:red">maximum 10 digit</font></td>
        </tr>
          <tr>
          <td colspan="2" align='center'><input type="submit"
value="submit" name="submit" /></td>
        </tr>
        </table>
</form>
         </center>
</body>
</html>
<%@ page language="java" contentType="text/html; charset=ISO-
8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="java.sql.*,java.util.*"%>
<%
String use=request.getParameter("use");
String pass=request.getParameter("pass");
String name=request.getParameter("name");
String ph_no=request.getParameter("ph_no");
try
{
Class.forName("org.apache.derby.jdbc.ClientDriver");
Connection conn=
DriverManager.getConnection("jdbc:derby://localhost:1527/demo",
"demo","demo");
Statement st=conn.createStatement();
int i=st.executeUpdate("insert into faculty
values('"+use+"','"+pass+"','"+name+"','"+ph_no+"')");
if(i>0)
out.println("<br>Thank you for register ! Please<br> <a
href='facultylogin.jsp'>Login</a> to continue.");
}
catch(Exception e)
{
System.out.print(e);
e.printStackTrace();
}
%>
```

- **Login-Admin**

```
<html>
<head>
<body background="ssimg.jpg" >
<form action="adminlogin.jsp" method="post" >
<center> <h3 style="background-color: pink;">ADMIN LOGIN
FORM</h3></center>
<TABLE border=0 cellspacing=2 cellpadding=2 width="100%">
<TR>
<TD width="34%">
<TABLE border=0 cellspacing=2 cellpadding=2 width="100%">
<TR>
<TD>
</TD>
</TR>
</TABLE>
</TD>
<TD>
   <table cellpadding=2 cellspacing=0 border=0>
<tr> <td bgcolor="red"><table cellpadding=0 cellspacing=0
border=1 width=100%> <tr> <td bgcolor="green" align=center
style="padding:2;padding-bottom:4"> <b><font size=-1
color="white" face="verdana,arial"> LOG IN</font></b></tr>
<tr><td bgcolor="white" style="padding:5"><br>
<TABLE border=0 cellspacing=5 cellpadding=2 width="60%"
bgcolor="white">
<TR>
<TD>
<tr><td> <font face="verdana,arial" size=-1>
Login_Type:</font></td> <td><select name="type"><option
value="admin">admin</option></select></td></tr>
```

```
<tr><td> <font face="verdana,arial" size=-1> Login:</font></td>
<td><input type="text" name="login" required
value=""></td></tr>
<tr><td> <font face="verdana,arial" size=-1>
Password:</font></td> <td><input type="password"
name="password1" required  ></td></tr>
<tr><td> <font face="verdana,arial" size=-1>&nbsp;</font></td>
<td><font face="verdana,arial" size=-1><input type="submit"
value="Enter"></font></td></tr>
<tr><td colspan=2><font face="verdana,arial" size=-
1>&nbsp;</font></td> </tr>
</TABLE>
</TD>
<TD>
</table>
</TD>
</TR>
</TABLE>
</TD>
</TR>
</TABLE>
<%
 String username=request.getParameter("login");
 String password=request.getParameter("password1");
if((username.equals("root") && password.equals("root")))
{ session.setAttribute("login",username);
response.sendRedirect("adminhome.html");
}
else
    //response.sendRedirect("Error.jsp");
%>
</form>
</body>
</html>
```

- **Logout-Admin**

```
<html>
        <SCRIPT type="text/javascript">
 window.history.forward();
 function noBack() { window.history.forward(); }
</SCRIPT>
<BODY bgcolor="khaki" onload="noBack();"
 onpageshow="if (event.persisted) noBack();" onunload="">
        <%
```

```
            session.invalidate();
          %>
        <center>
            <h1><font color="Red">You are Sucessfully logged
out...</font></h1>
            <a href="newjsp.jsp">Go-Back To Home Page</a>
        </center>
        </body>
</html>
```

- **Download Files**

```
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.OutputStream;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(name = "DownloadServlet", urlPatterns =
{"/DownloadServlet"})
public class DownloadServlet extends HttpServlet {
    public static int BUFFER_SIZE = 1024 * 100;
    public static final String UPLOAD_DIR = "resources";
    public static String fileName = null;
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        /**
         * *** Get The Absolute Path Of The File To Be Downloaded
****
         */
        fileName = request.getParameter("fileName");
        if (fileName == null || fileName.equals("")) {
            /**
             * *** Set Response Content Type ****
             */
            response.setContentType("text/html");
            /**
             * *** Print The Response ****
             */
            response.getWriter().println("<h3>File " + fileName + " Is
Not Present .....!</h3>");
        } else {
            String applicationPath =
getServletContext().getRealPath("");
            String downloadPath = applicationPath + File.separator +
UPLOAD_DIR;
            String filePath = downloadPath + File.separator + fileName;
            System.out.println(fileName);
            System.out.println(filePath);
            System.out.println("fileName:" + fileName);
            System.out.println("filePath :" + filePath);
            File file = new File(filePath);
            OutputStream outStream = null;
            FileInputStream inputStream = null;
            if (file.exists()) {
                /**
                 * ** Setting The Content Attributes For The Response
Object
                 * ***
                 */
                String mimeType = "application/octet-stream";
                response.setContentType(mimeType);
                /**
                 * ** Setting The Headers For The Response Object ***
                 */
                String headerKey = "Content-Disposition";
                String headerValue = String.format("attachment;
filename=\"%s\"", file.getName());
                response.setHeader(headerKey, headerValue);
                try {
                    /**
                     * ** Get The Output Stream Of The Response ***
                     */
                    outStream = response.getOutputStream();
                    inputStream = new FileInputStream(file);
                    byte[] buffer = new byte[BUFFER_SIZE];
                    int bytesRead = -1;
                    /**
                     * ** Write Each Byte Of Data Read From The Input
Stream
                     * Write Each Byte Of Data Read From The Input
Stream Into
                     * The Output Stream ***
                     */
                    while ((bytesRead = inputStream.read(buffer)) != -1) {
                        outStream.write(buffer, 0, bytesRead);
                    }
                } catch (IOException ioExObj) {
                    System.out.println("Exception While Performing The
I/O Operation?= " + ioExObj.getMessage());
                } finally {
                    if (inputStream != null) {
                        inputStream.close();
                    }
                    outStream.flush();
                    if (outStream != null) {
                        outStream.close();
                    }
                }
            } else {
                /**
                 * *** Set Response Content Type ****
                 */
                response.setContentType("text/html");
                /**
                 * *** Print The Response ****
                 */
                response.getWriter().println("<h3>File " + fileName + " Is
Not Present .....!</h3>");
            }
        }
    }
}
```

Testing

- **Unit Tests**

Unit tests are most commonly done by developers on their own machines or on a common server that is very volatile. It is not necessary that the unit test machines be the same platform and operating system as the target deployment environment, but the movement from the unit test environment to other testing environments should not require material code changes by developers. A plan for one machine per developer plus one small server should be included in the overall system architecture.

- **System Tests**

The system test environment allows multiple modules to be connected together and executed as in a typical use-case scenario. The choice as to whether this is done on a separate machine from unit testing is up to the implementation and test team. If the target deployment environment is different from the unit test environment, the system test environment should contain a machine that matches the target environment. Although the system test machine need not match the size of the deployment box, it should have the same platform and operating system. A good rule of thumb is to prepare to add one more box for system tests of a smaller size, but the same operating system as the target environment. Again, this will be a relatively volatile environment, so it should not be viewed as a place to do industrial-strength testing by a large team.

- **Integration or Regression Tests**

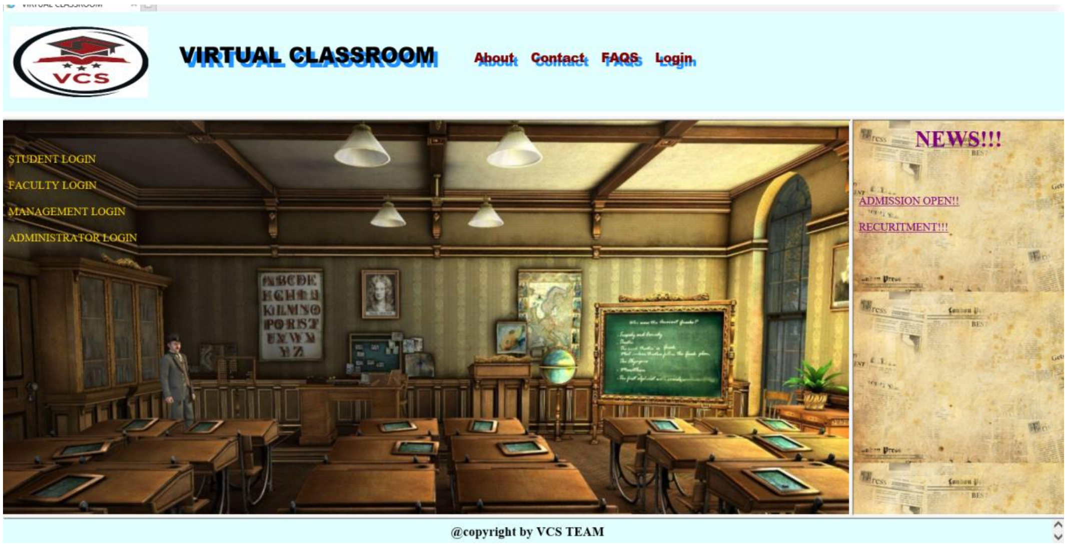
To perform integration and regression tests, it is advisable to have a separate environment that is similar to the target environment. Generally, one server will be enough at this point. However, the contents of this server should be strictly controlled. Either the test coordinator or his or her designate should make all software changes to this environment. Stability and auditability are essential to ensuring the accuracy of test results. Plan for at least one more servers at this stage in testing.

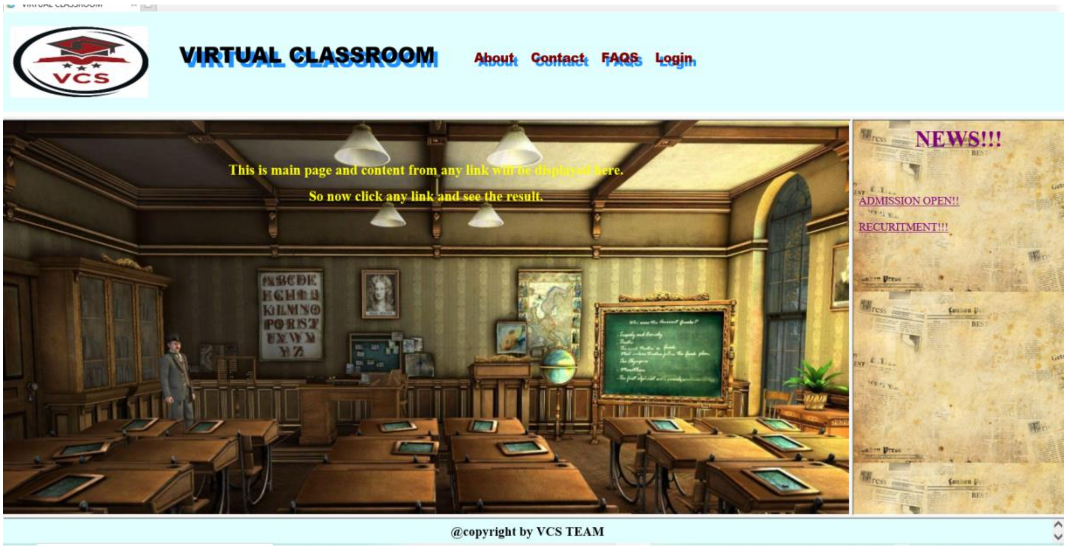
- **Acceptance Tests**

Acceptance tests are generally performed in the same environment as the stress tests, so additional hardware is not needed to support this phase of testing. Again, during the initial development cycle, the production environment can be used to perform both acceptance testing and staging, but a new environment should be created for subsequent development cycles.

User Interfaces

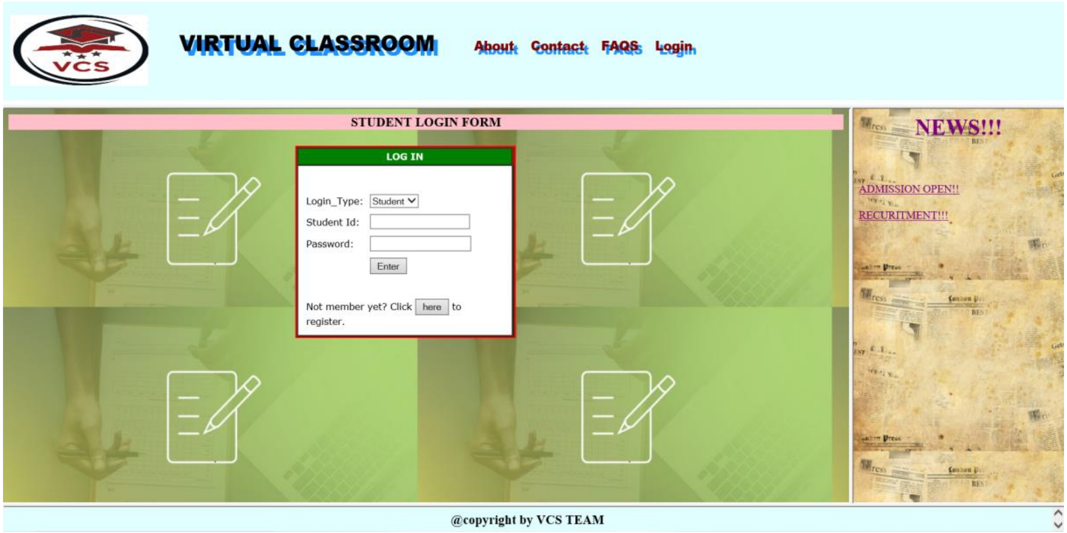
- **Screen Images**





This is main page and content from any link will be displayed here.

So now click any link and see the result.



STUDENT LOGIN FORM

LOG IN

Login_Type: Student

Student Id:

Password:

Enter

Not member yet? Click here to register.

Appendices

VCS: Virtual Classroom System
Req: Request

References

- http://findarticles.com/
- http://en.wikipedia.org/wiki/History_of_virtual_learning_environments/
- http://ezinearticles.com/
- IEEE SRS Format Std 830-1998
- Software Engineering by Ian Sommerville
- Writing Software requirement specification by Donn Le Vie, Jr.