वनस्थली विद्यापीठ
Banasthali University
University for Women : University with a difference

Department of Computer Science and Mathematics
Banasthali University

A project report on

Admit Card Automation System

Group ID: A1

Project Coordinators:
Mrs.Dipanwita Thakur
Dr.Deepak Kumar

Project Mentor:
Mr.Manjeet Kumar

Submitted by:
Anjali Deshwani (9045)
Ayushi Gaur (9065)
Chaya Mishra (9073)
Deepti Dixit (9077)
B.Tech (CS) 3rd year
(VI Semester)

<u>**CERTIFICATE**</u>

This is to certify that Anjali Deshwani, Ayushi Gaur, Chaya Mishra and Deepti Dixit have worked for the project entitled "**Admit Card Automation System"** at Banasthali University under my supervision and this is a bonafide piece of their work. This original project was undertaken by them as a part of partial fulfillment of the requirement for the award of degree of Bachelors of Technology (B.Tech).
The work done in the project is the result of their own efforts, creativity & ideas.

Dr.Sarla Pareek	**Mr. Manjeet Kumar**
Dean, AIM & ACT	(Project Mentor)
Banasthali University	

Abstract

This project is a web application for the students of the University which would allow them to apply for the admit card of the semester exams. It would also reduce the manual work of the University officials by automating the verification procedure i.e. verifying admit card form details with the student details present in the University database.

The scope of the project is within the boundaries of Banasthali Vidyapith. It can be used by students and employees of the University. In the existing system in the University, the students apply for the admit card manually which is tedious for the students. The officials also have to verify the forms manually in the present scenario. This project aims at reducing the workload of the students as well as the employees by automating the process.This project provides a user friendly interface for students to apply for admit card.

The report includes Software Requirement Specification which illustrates the purpose and gives a complete declaration for the development of system. It explains the system constraints, interfaces and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

This report also includes Software Design Specification which shows how the software will be structured to satisfy the requirements identified in the SRS. It is a translation of requirements into a description of software structure, software components, interfaces and data necessary for the implementation phase. It provides the design details of admit card automation system. The expected audience is administrator, students, University officials and any visitor of the website.

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude towards all those people who in various ways have helped in the successful completion of our project.

Firstly, we would like to thank our mentor Mr Manjeet Kumar who gave us his valuable suggestions and ideas when we were in need of them. He encouraged us to work on this project.

We are also greatful to our college and our project coordinators Mrs.Dipanwita Thakur and Dr. Deepak Kumar for giving us the opportunity to work with them and providing us necessary resources for the project.

We are also thankful to the officials of Vidya Mandir for providing the detailed procedure of Admit card generation. We'd like to thank technical assistant of AIM &ACT Department for their cooperation throughout the project work. We hereby offer our sincere compliment to all our friends for their useful suggestion and cooperation.

We are immensely grateful to all involved in this project as without their inspiration and valuable suggestion it would not have been possible to develop the project within the prescribed time.

TABLE OF CONTENTS

1.OBJECTIVE

The scope of the project is within the boundaries of Banasthali Vidyapith. It can be used by students and employees of the University. It is the web application for admit card generationfor the students.In the existing system in the University, the studentsapply for the admit card manuallywhich is tedious for the students. The officials also have to verify the forms manually in the present scenario.This project aims at reducing the workload of the students as well as the employees by automating the process.

The scope of the project are enumerated as under:

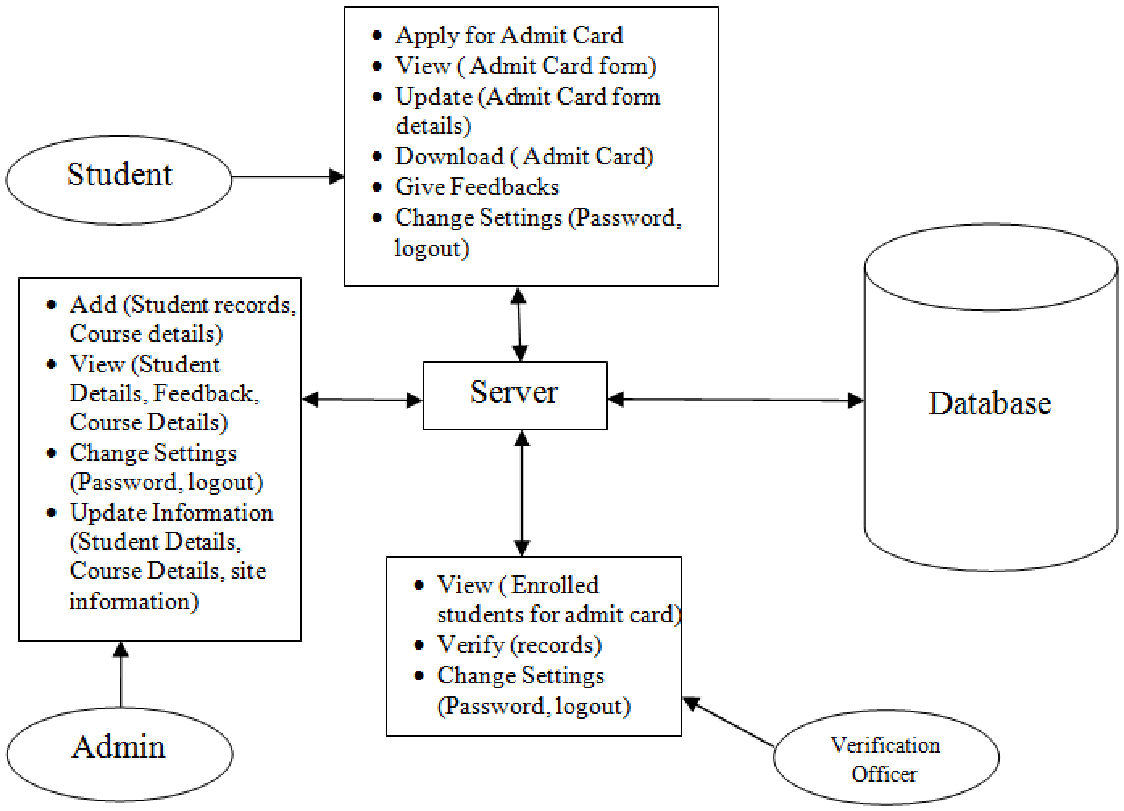
- It has been designed keeping in view the common problems faced by the students and the employees of the University.
- It assists the officials in verification of the admit card form.
- By accepting the details from the student, it generates their admit card.
- Along with generation of admit cards, it also allows them to update their information.

2.SOFTWARE REQUIREMENT SPECIFICATION

2.1 Product Perspective

This is a web application for the students of the University which would allow them to apply for the admit card of the semester exams. It would also reduce the manual work of the University officials by automating the verification procedure i.e. verifying admit card form details with the student details present in the University database.

This product requires databases to store the student details and the admit card details.


Student
Apply for Admit Card
View (Admit Card form)
Update (Admit Card form details)
Download (Admit Card)
Give Feedbacks
Change Settings (Password, logout)
Add (Student records, Course details)
View (Student Details, Feedback, Course Details)
Change Settings (Password, logout)
Update Information (Student Details, Course Details, site information)
Server
Database
View (Enrolled students for admit card)
Verify (records)
Change Settings (Password, logout)
Admin
Verification Officer

Block Diagram to show various components of the system

2.2 Hardware Requirements

Server Side

Processor – 1-2 GHz (ex .P-IV) or onwards

RAM - 512 MB

HDD – 20GB or more (free space excluding data size)

Client Side

Processor – 450MHz (ex. P-II) or onwards

RAM – 128 MB

HDD – 10GB or more (free space excluding data size)

Developer Side

Processor – 2.50GHz Core i3

RAM – 4GB

HDD – 20GB or more (free space excluding data size)

2.3 Software Requirements

Server Side

OS - Any Operating System

Web Server –Apache Tomcat (any version)

Database Server – MySQL server 5.0 onwards

Client Side

OS – Any operating system

Browser – Any browser compatible with IE 5.0 or onwards

Developer side

OS – Windows 8.1/10

Front End –JSP

Database – MySQL server 5.0

Scripting Language- JavaScript, CSS

Browser– Any browser compatible with IE 5.0 or onwards

Server – Apache Tomcat (any version)

Tools- NetBeans 8.1/8.2

Technologies used:

Programming Langugage:Java

Technology: Netbeans

Web Server: Glassfish 4.0

Script Language: JavaScript

2.4 Product Functions

2.4.1 System Interface

The product will have a simple interface; complete with message communication and indicators for all available options. The footer of the main page will carry the copyright information and names of web application developers.

There will be a menu on each page that will provide access to the following screens:

- **About us:** This page will provide information about the web application development team. It will display the names of team members by default, but will have the option of viewing details.
- **Student:** The students will be directed to a default page which will provide links to the following -

    - Login: A login area will be provided where user will be required to supply username and password in order to log into his account and listen to songs.
    - Signup: The sign-up facility will be provided to the user in order to create an account and hence become a registered user. The information required will include username, password, confirm password, email-id, date of birth, security question and answer.
- **Verification Officer :**The user will be directed to a default page which will provide links to the following -
    - Login: A login area will be provided where user will be required to supply username and password in order to log into his account and listen to songs.
    - Signup: The sign-up facility will be provided to the user in order to create an account and hence become a registered user. The information required will include username, password, confirm password, email-id, date of birth, security question and answer.
- **Admin:** The admin will be directed to a default page which will provide links to the following -
    - Login: A login area will be provided where user will be required to supply username and password in order to log into his account and listen to songs.
    - Signup: The sign-up facility will be provided to the user in order to create an account and hence become a registered user. The information required will include username, password, confirm password, email-id, date of birth, security question and answer.
- **Guidelines for the students:** This provides steps for the students to apply for admit card.
- **FAQs:** This will display the frequently asked questions and their answers.
- **Feedback:** The user will be directed to this page on clicking the feedback option. It has a text area where the user can give his/her feedback.

- **Contact Us:** This page will simply display the contact information about the web application development team.

The following interfaces will be available:

To the students:

- **Default:**The User will be automatically directed to this screen when he/she logs in. User will be given options to view apply for admit card, view/update the admit card form and download the admit card once available.
    - Admit Card Apply: This link will allow the student to apply for the admit card by filling their personal and course details along with semester information.
    - View/Update Admit Card: This link will simply allow the students to view or update their admit card form.
    - Admit Card Download: This link will enable the students to download the admit card after being verified by the officer in-charge.
    - Feedback: The student will be directed to this page. It will have a text area where the user can type in his feedback. The student will have to specify her name and email-id.
    - Change Settings: This page will allow user to change the password. It will also give an option to Logout. After Logout, the user will be redirected to main page.

To the Verification Officer:

- **Default:** The User will be automatically directed to this screen when he/she logs in. User will be given options to view the students who have enrolled for admit card and to verify the records.
    - View:The officer can view the students who have applied for the admit card and the ones who have been verified.
    - Verify the admit card forms**:** The officer verifies the form details from the Student and Course records.

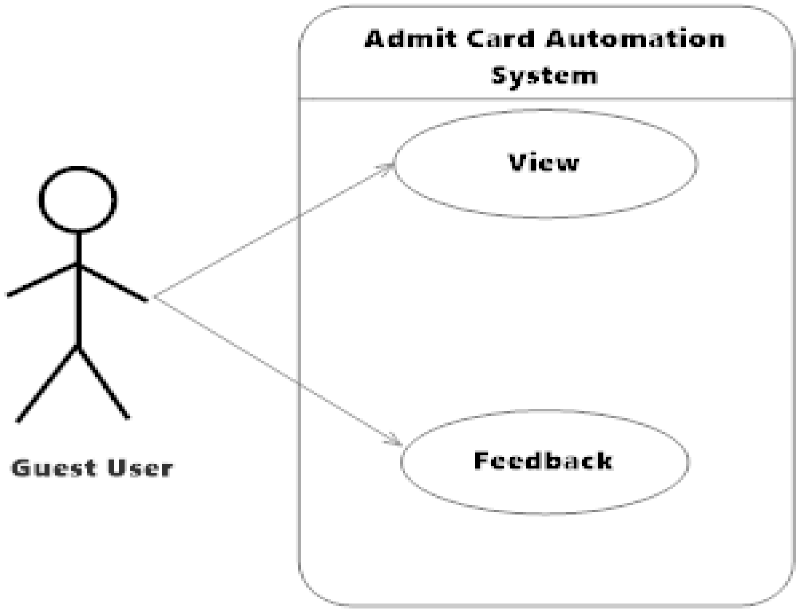
    - Change Settings: This page will allow user to change the password. It will also give an option to Logout. After Logout, the user will be redirected to main page.

To the Admin:

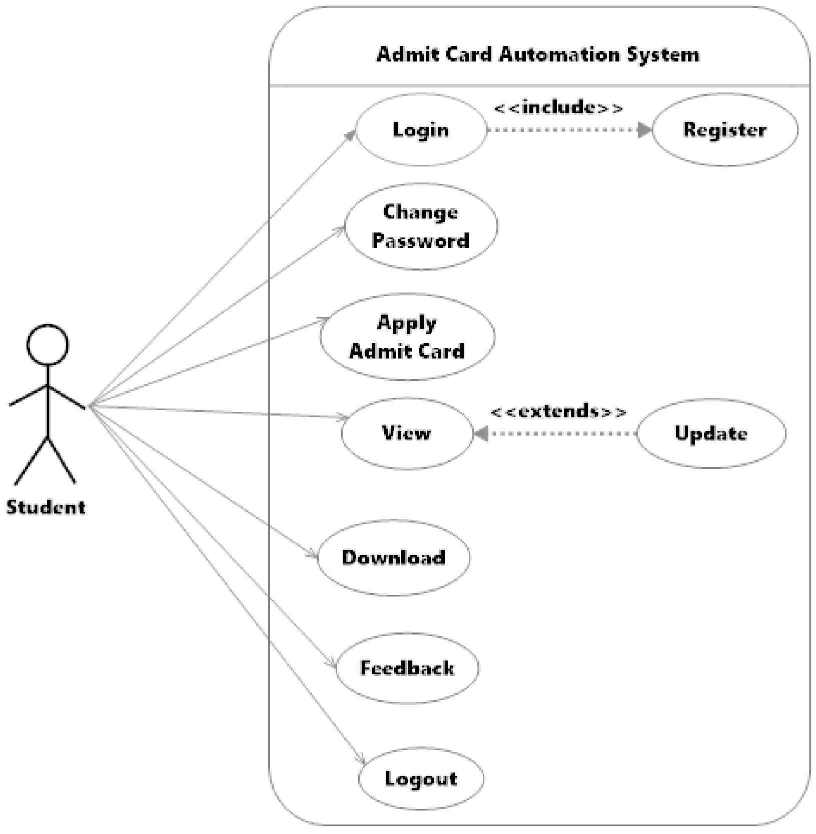
- **Default:** The User will be automatically directed to this screen when he/she logs in. User will be given options to add student records, update student records, add course subjects per semester, view registered students, view feedbacks and maintaining the portal.
    - Add Student/Course details: The admin can add new student records and courses introduced.
    - Update Student/Course details: The admin can modify student records and course details.
    - View: The admin can view the feedback and the list of students who have registered.
    - Change Settings: This page will allow user to change the password. It will also give an option to Logout. After Logout, the user will be redirected to main page.

2.5 Use Case Diagrams

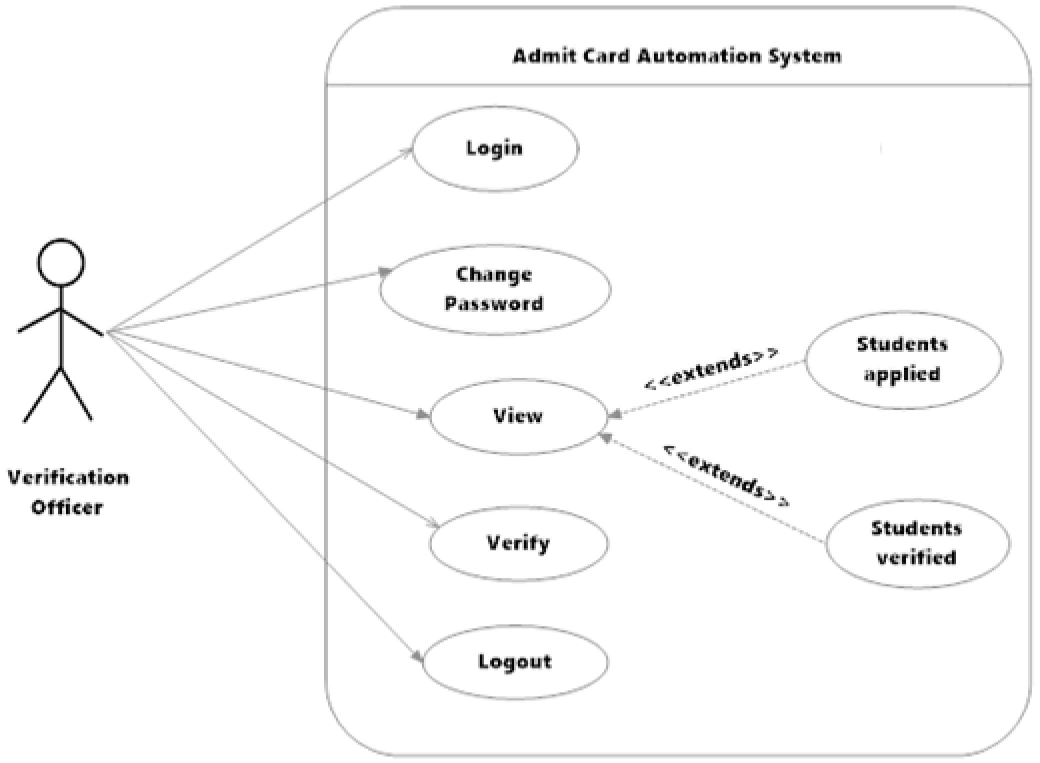
1. **Guest User**


Admit Card Automation System
View
Feedback
Guest User

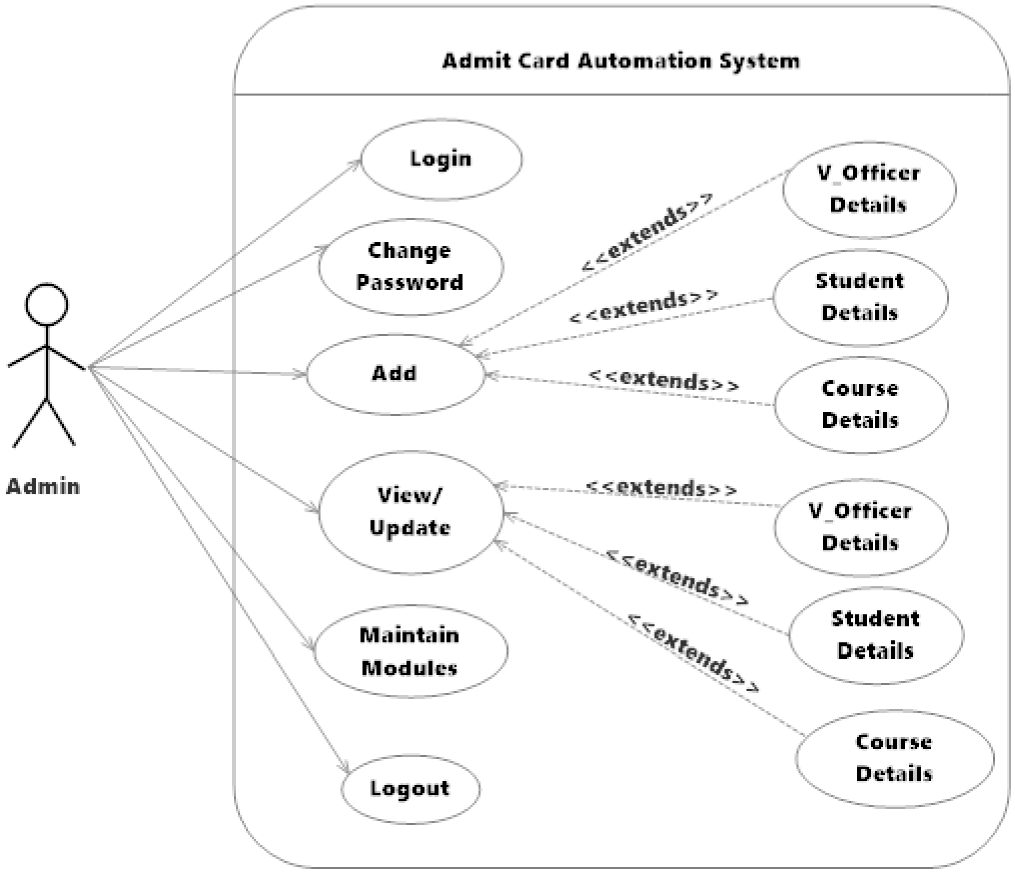
2. **Student**


Admit Card Automation System
Login
<<include>>
Register
Change Password
Apply Admit Card
View
<<extends>>
Update
Download
Feedback
Logout
Student

3. Verification Officer



4. **<u>Admin</u>**


Admit Card Automation System
Admin
Login
Change Password
Add
View/ Update
Maintain Modules
Logout
V_Officer Details
Student Details
Course Details
<<extends>>
V_Officer Details
Student Details
Course Details

2.6 Functional Specifications (Use Case Description)

Use Case 1: Login

USE CASE NO	1
USE CASE NAME	Login
ACTORS	Admin, Students, Verification Officer.
DESCRIPTION	Login
NECESSARY CONDITION	The users must be registered.
NORMAL COURSE EVENT	1) Actors enter their username 2) Actors enter their password 3) Actors click login button 4) System connects to database 5) Homepage displayed

Use Case 2 : Log Out

USE CASE NO	2
USE CASE NAME	Log Out
ACTORS	Student, Admin and Verification Officer
DESCRIPTION	This module helps the admin, the verification officer and students to log out.
NECESSARY CONDITION	The user must be logged in.
NORMAL COURSE EVENT	1) The application user clicks on the ‘log out’ button. 2) Database connection terminated. 3) The user logs out successfully.
OUTPUT	Application homepage will be displayed.

Use Case 3 : Password Change

USE CASE NO	3
USE CASE NAME	Password Change
ACTORS	Student, Admin and Verification Officer
DESCRIPTION	This module helps the admin, the verification officer and students to change their password.
NECESSARY CONDITION	1) The user must be a member of the system. 2) The user must be logged in.
INPUT	1) Username 2) Previous Password
NORMAL COURSE EVENT	1) The application user clicks on the ‘change password’ button. 2) User enters the new password which the user wants. 3) Database connection is initiated. 4) Password is updated into the database.
OUTPUT	User’s login page will be displayed.

Use Case 4: Student’s view pages

USE CASE NO	4
USE CASE NAME	Student’s view pages
ACTORS	Students
DESCRIPTION	Various modules of the system that the student can view.
NECESSARY CONDITION	Filled Admit Card Form 1) Student must be logged in. 2) Student must have filled the form.
NORMAL COURSE EVENT	1) Student clicks on the ‘view form’ button. 2) Database connection is initiated.
OUTPUT	Student’s filled form is displayed.

Use Case 5 - Admin’s View Pages

USE CASE NO	5
USE CASE NAME	Admin’s view pages

ACTORS	Admin
DESCRIPTION	Various modules of the system that the admin can view.
NECESSARY CONDITION	<u>Student Details</u> 1) Admin must be logged in. 2) Proper database of student details must be maintained. <u>Course Details</u> 1) Admin must have logged in. 2) Proper database of course details must be maintained.
NORMAL COURSE EVENT	1) Admin clicks on ‘Student Details’ / ‘Course Details’ button. 2) Database connection is initiated.
OUTPUT	Details of all the enrolled students / courses are displayed.

<u>Use Case 6 : Verification Officer’s view pages</u>

USE CASE NO	6
USE CASE NAME	Verification Officer’s view pages
ACTORS	Verification Officer
DESCRIPTION	Various modules of the system that the Verification Officer can view.
NECESSARY CONDITION	<u>Applied Students</u> 1) Verification Officer must be logged in. 2) Students must have filled the form. <u>Verified Students</u> Verification Officer must have logged in.
NORMAL COURSE EVENT	1) Verification Officer clicks on ‘verified students’ or applied students’ button. 2) Database connection is initiated.
OUTPUT	List of all applied students / verified students is displayed.

<u>Use case 7 : Update</u>

USE CASE NO.	7
USE CASE NAME	Admit card details Updation.
ACTORS	Student
DESCRIPTION	This Module helps the Student to Update the information about themselves in registered form.
NECESSARY CONDITION	The user must be logged on to the system.

INPUT	Information to be modified.
NORMAL COURSE EVENT	1) The user must be logged on to the system which is defined in use case. 2) The user taps on Update Registered form. 3) The system retrieves user information from DB. 4) The user can update his/her form as in desired information field. 5) The user taps on Submit Button. 6) The user details are updated in the desired database.
ALTERNATE COURSE EVENT	1) The system puts a message on the top of the window about the problem. 2) Continue with step 2 in the normal course events.
OUTPUT	User information will be updated.

Use Case 8: Update

USE CASE NO.	8
USE CASE NAME	Update Course and Student details
ACTORS	Admin
DESCRIPTION	This module helps to admin to Update detail of course and student in desired database
NECESSARY CONDITION	The Admin must be logged to the system.
INPUT	Information to be modified.
NORMAL COURSE EVENT	1) The admin must be logged on to the system which is defined in use case 1. 2) The admin taps on Update Profile. 3) The system retrieves user information from DB. 4) The admin can update profiles of desired student and verification Officer informatio field. 5) The admin taps on Save Button. 6) The user profile is saved in desired database.
OUTPUT	Information related to student and verification officer will get updated in desired database.

Use case-9 Registration

USE CASE NO.	9
USE CASE NAME	Registration.
ACTORS	Admin, Verification Officer, Student

DESCRIPTION	Users SignUp.
NECESSARY CONDITION	New user with new Smart Card ID is required.
NORMAL COURSE EVENT	1) Actors enter their smart card id, Aadhaar Card 2) Actors enter their password 3) Actors enter required information 4) Actor tap register button 5) System connects to database 6) A message appears which shows that they are registered.
ALTERNATIVE COURSE EVENT	1) Same smart card id exists in database 2) Error message will appear 3) Continue with step1 in normal course events 4) An error may occur during database operation 5) System show error messages

<u>Use case 10 : Apply Admit Card</u>

USE CASE NO.	10
USE CASE NAME	Apply Admit Card
ACTORS	Student
DESCRIPTION	Apply for admit card.
NECESSARY CONDITION	The users must login.
NORMAL COURSE EVENT	1) click the apply button 2) Actors are redirected to the form page. 3) Actors must fill the form. 4) Actors click submit button. 5) The details are submitted to system database.
OUTPUT	The form details are submitted to system database.

<u>Use case 11 :Download</u>

USE CASE NO.	11
USE CASE NAME	Download
ACTORS	Student
DESCRIPTION	Download the verified admit card.
NECESSARY CONDITION	The users must login.
NORMAL COURSE EVENT	1) Actors click the download button. 2) Actors are redirected to the page where they view their admit card. 3) Actors can save the admit card.

ALTERNATIVE COURSE EVENT	1) Actors click the download button. 2) The admit card is not yet verified and hence is not available for download. 3) System displays the message.
OUTPUT	Student downloads the admit card.

Use case 12 : Verify

USE CASE NO.	12
USE CASE NAME	Verify
ACTORS	Verification Officer
DESCRIPTION	Verify the admit card form filled by the student with the Student records.
NECESSARY CONDITION	The users must login.
NORMAL COURSE EVENT	1) Actors click the verify button. 2) Actors are redirected to the page where they can view the list of students who have applied for the admit card. 3) Actors verify the form details with that present in the database.
ALTERNATIVE COURSE EVENT	1) Actors click the verify button. 2) Student would not have filled the form. 3) Actors wait for student to apply for admit card.
OUTPUT	Verification Officer verifies the admit card.

Use case 13 : Feedback

USE CASE NO.	13
USE CASE NAME	Feedback
ACTORS	Student
DESCRIPTION	Give feedback.
NECESSARY CONDITION	The users must login.
NORMAL COURSE EVENT	1) Actors click the feedback option. 2) Actors are redirected to the page where they can fill the feedback. 3) Actors enter their name and feedback in description box. 4) Actors click submit button. 5) The feedback is recorded in the database.

OUTPUT	The feedback is saved in the database.

Use case 14 :Maintain modules

USE CASE NO.	14
USE CASE NAME	Maintain modules
ACTORS	Admin
DESCRIPTION	Maintains the site by updating the site content like About Us, Guidelines, Contact information.
NECESSARY CONDITION	The users must login.
NORMAL COURSE EVENT	1) Actors click the maintain modules option. 2) Actors are redirected to the page where they view different options to update like About Us, Guidelines, FAQs, Contact Us information. 3) Actors choose the option they want to update. 4) Actors update the details of chosen option. 5) Actor clicks submit button. 6) The changes are made and reflected on the site.
OUTPUT	Admin updates the respective module.

2.7 Software System Attributes

There are a number of quality attributes of software that can serve as requirements. It is important to specify required attributes so that their achievements can be objectively verified.

2.7.1 RELIABILITY

The system should work reliably, with automatic backup and recovery features. In case of unexpected termination of a session, the unsaved data should be recovered without loss and displayed to the respective users for saving into the system or continuing with the work.

2.7.2 AVAILABILITY

The entire system should be available round the year, except for a periodic maintenance. The maintenance period should be pre scheduled and short. The users should be reminded of the unavailability period, well in advance.

2.7.3 SECURITY

The system, at any time, should be accessed only by the authenticated users. The system has different modules which have been assigned certain functions, thus providing data abstraction and encapsulation.

2.7.4 MAINTAINABILITY

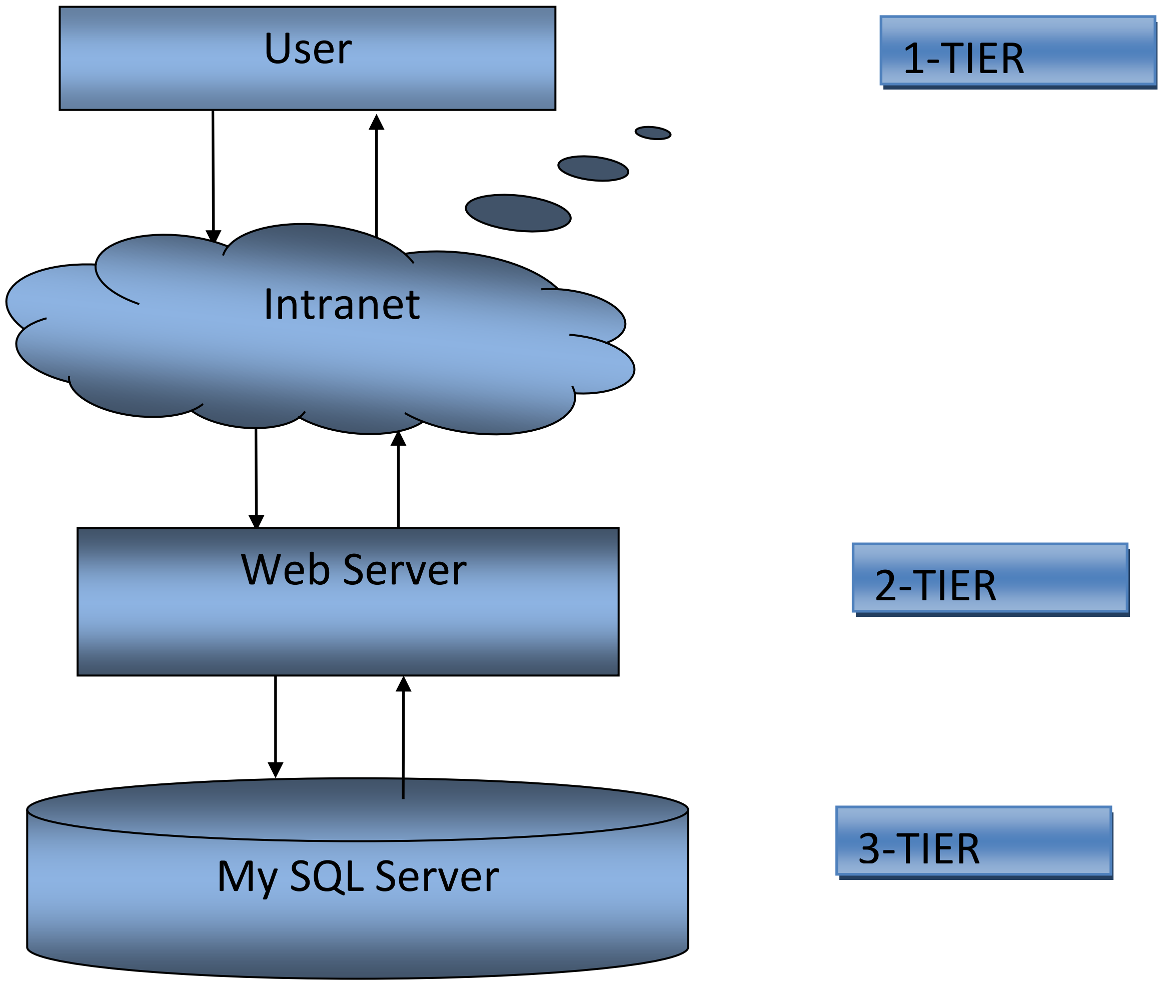
The maintainability shall be easily done by integrating new modules and updating the existing modules at specific intervals of times.

2.7.5 PORATBILITY

The system should support new versions of the related browsers. The administrative and server technologies should be standard and supported by most platforms.

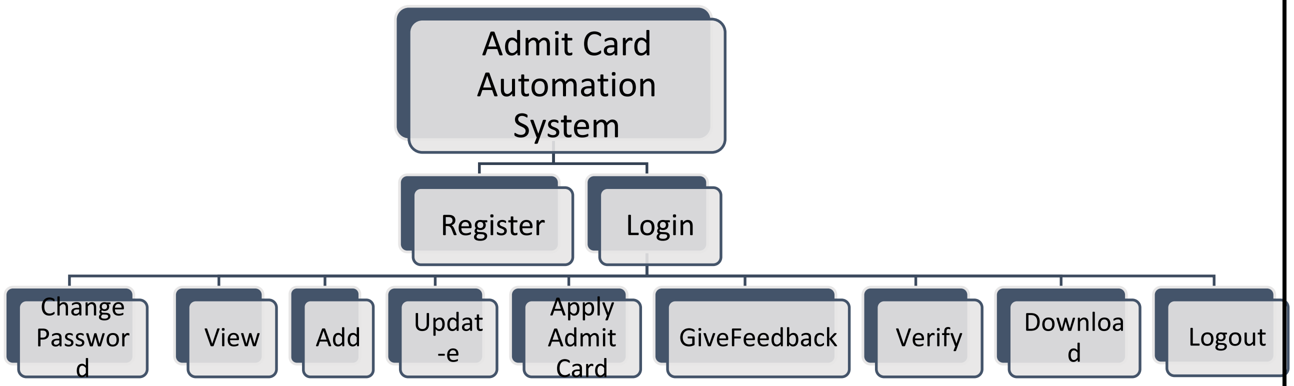
3.SOFTWARE DESIGN SPECIFICATION

3.1 High Level Diagram Overview



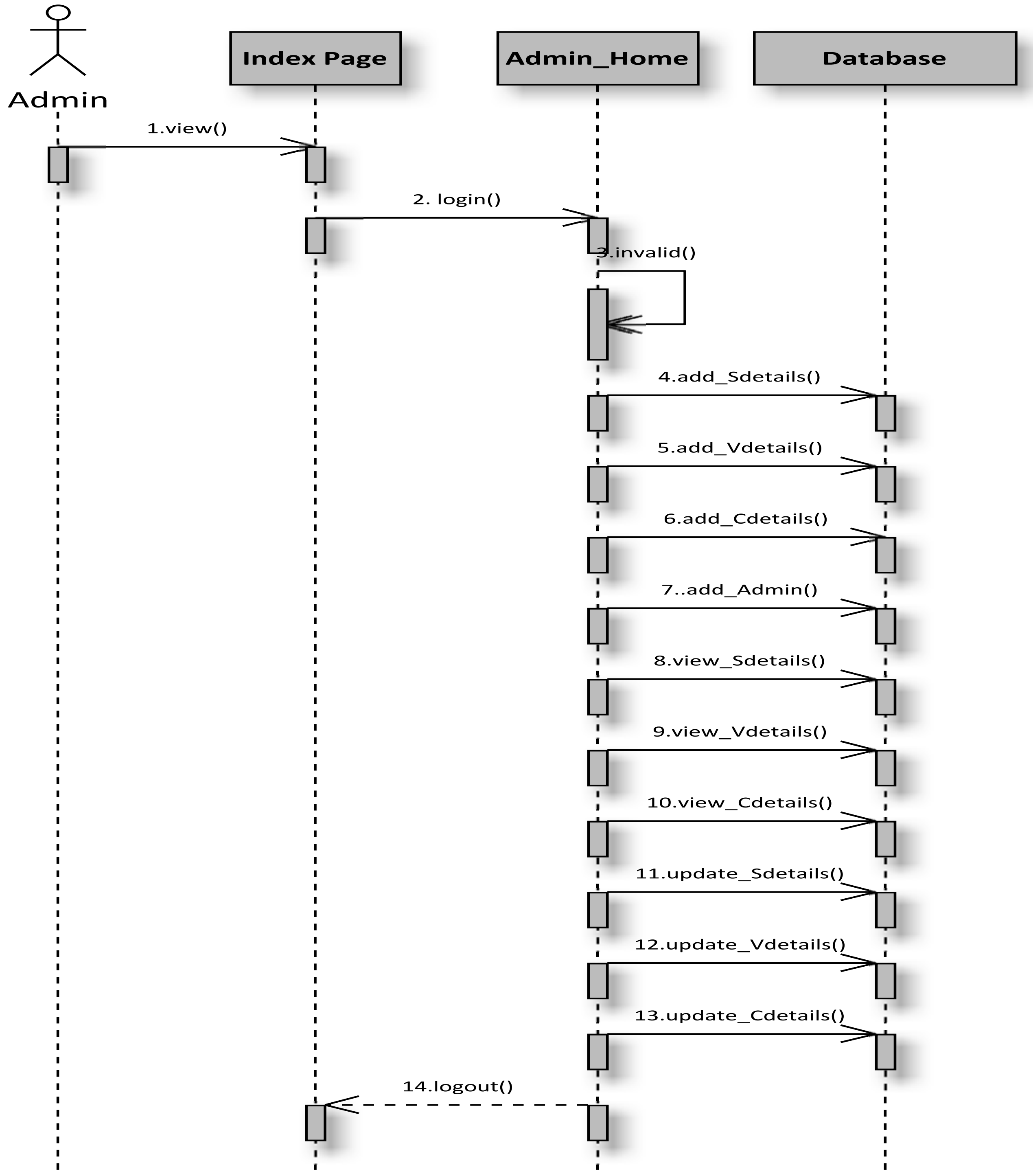
3.2 Detailed Description of Components

3.2.1 Work Breakdown Diagram

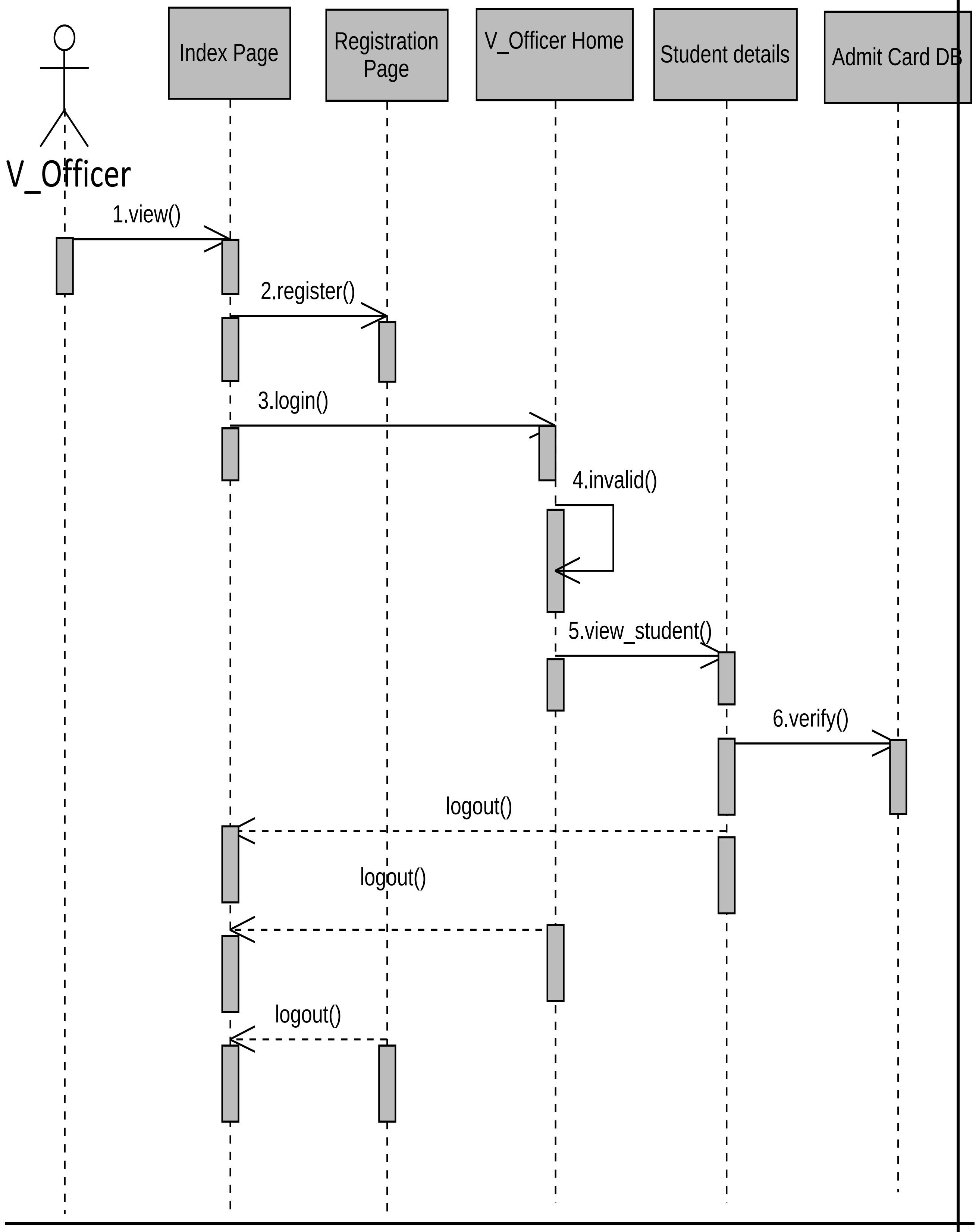


3.2.2 Sequence Diagram

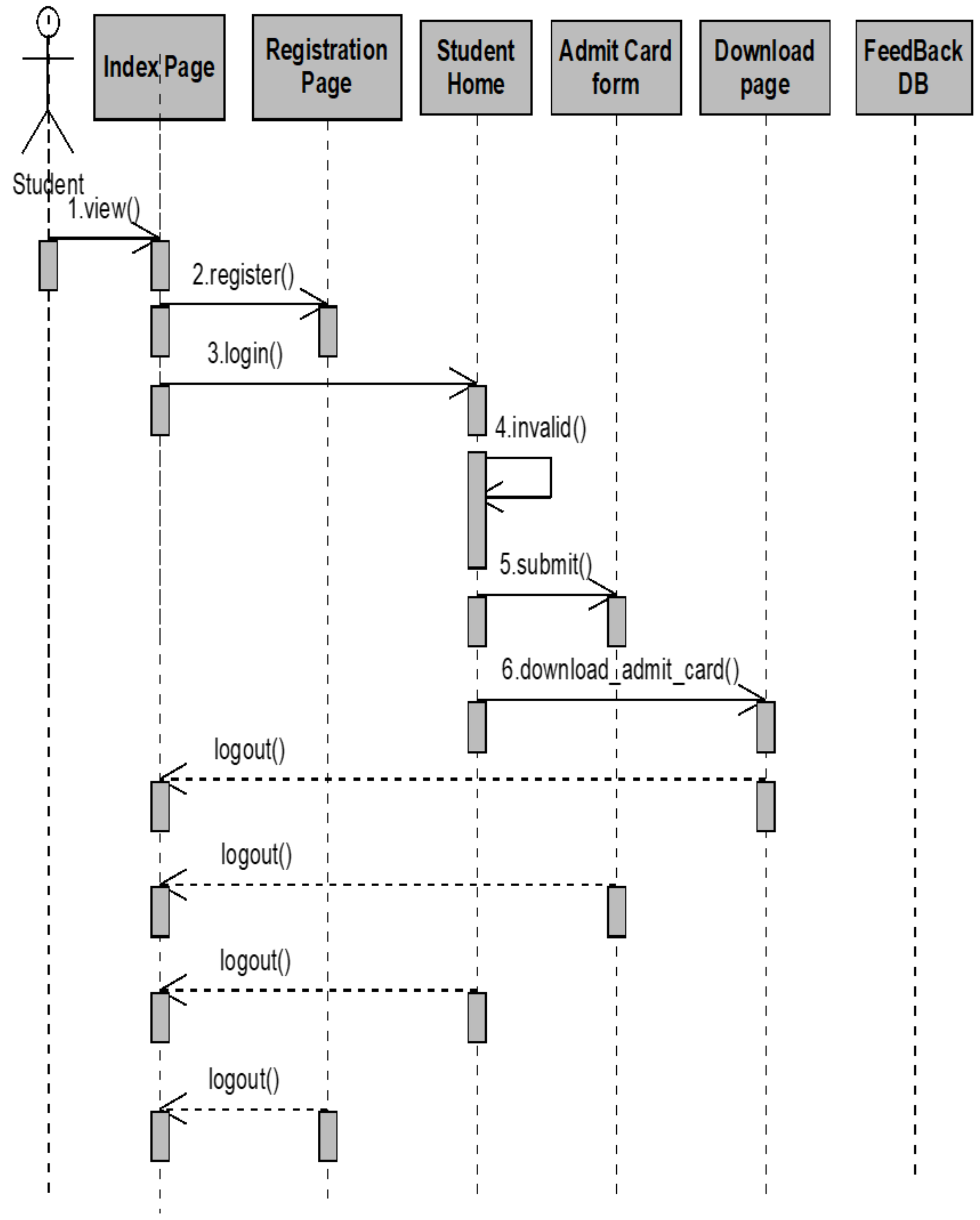
1.Admin



2.Verification Officer

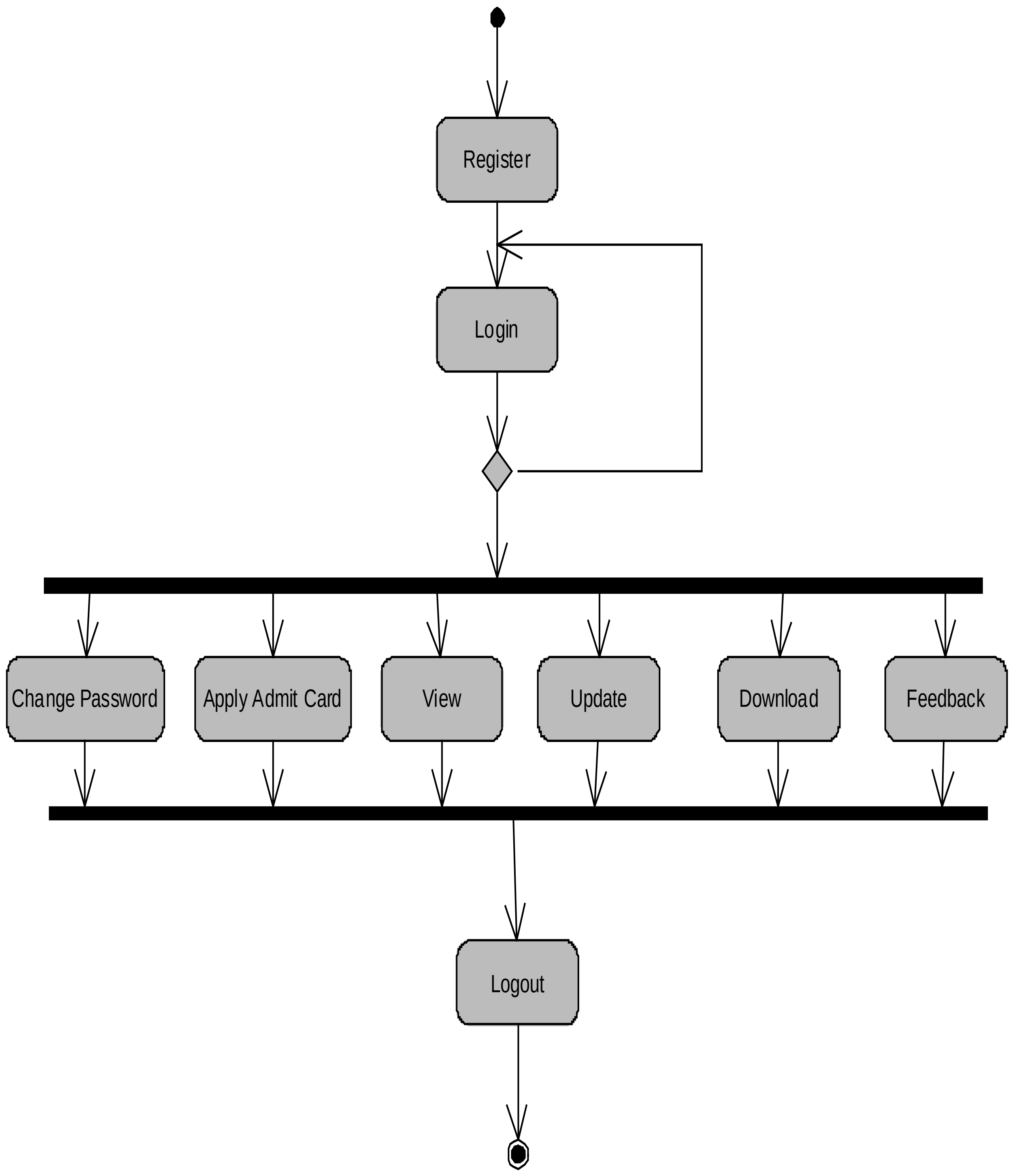

V_Officer
Index Page
Registration Page
V_Officer Home
Student details
Admit Card DB
1.view()
2.register()
3.login()
4.invalid()
5.view_student()
6.verify()
logout()
logout()
logout()

3.Student

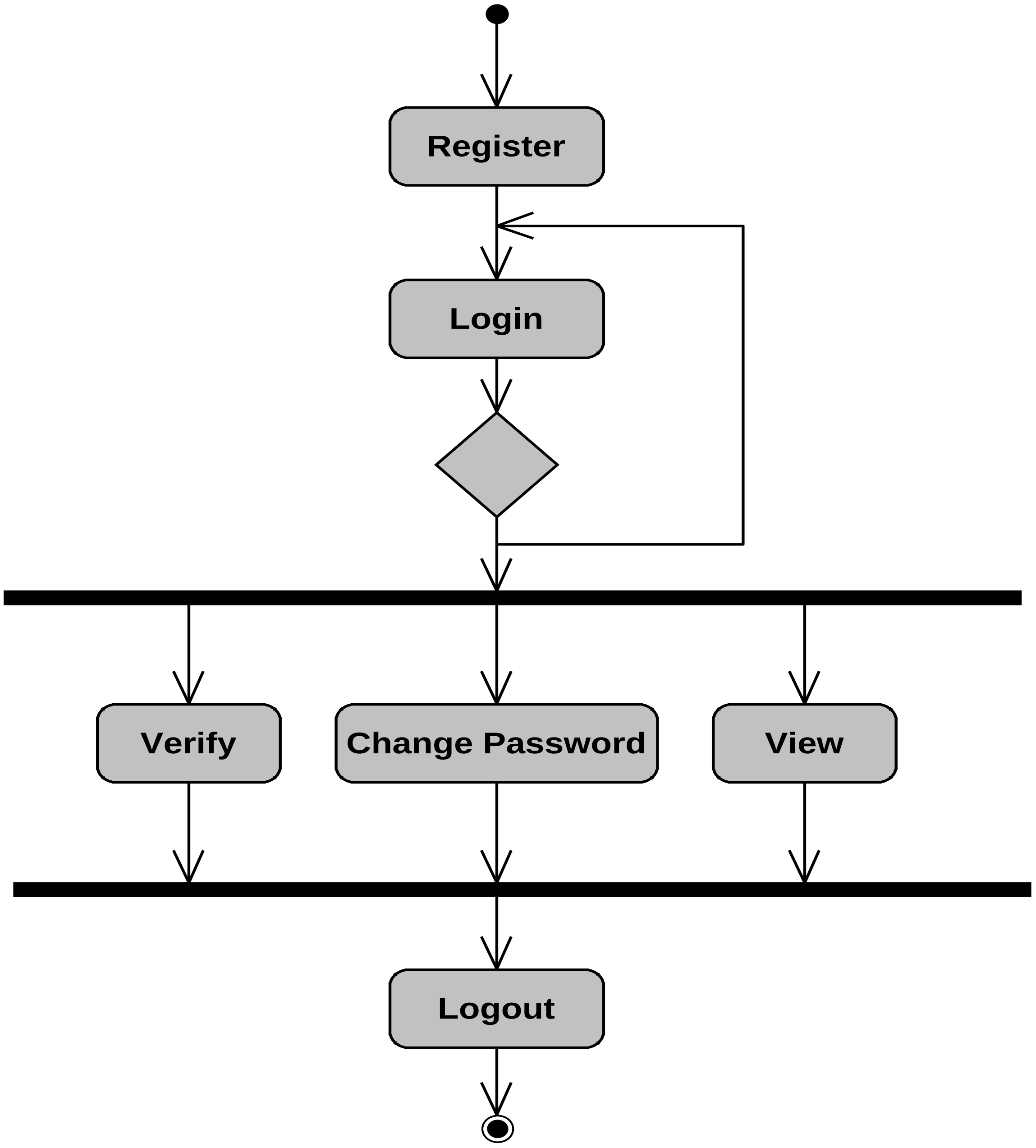


3.2.3 Activity Diagram

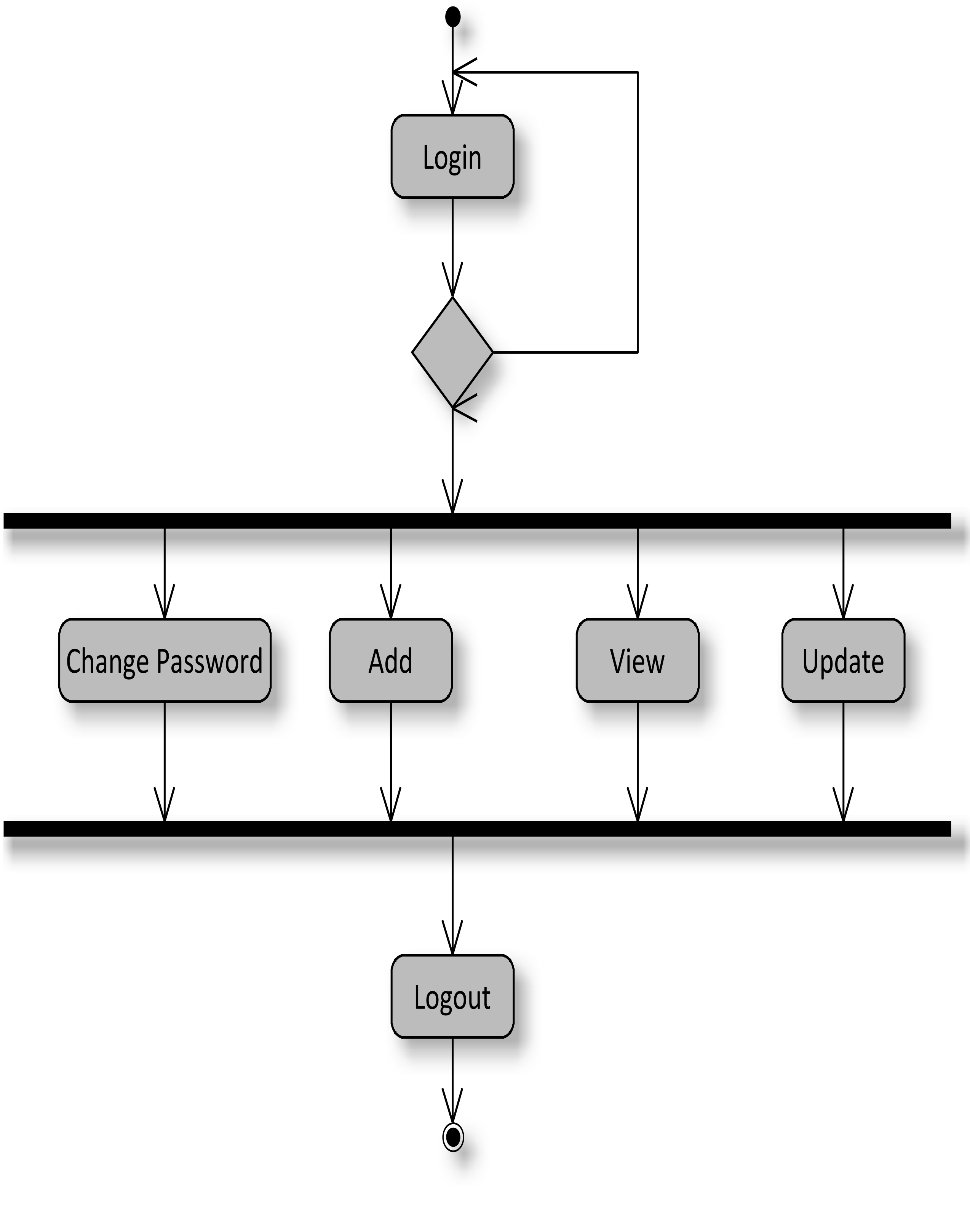
1. **Student**


Register
Login
Change Password
Apply Admit Card
View
Update
Download
Feedback
Logout

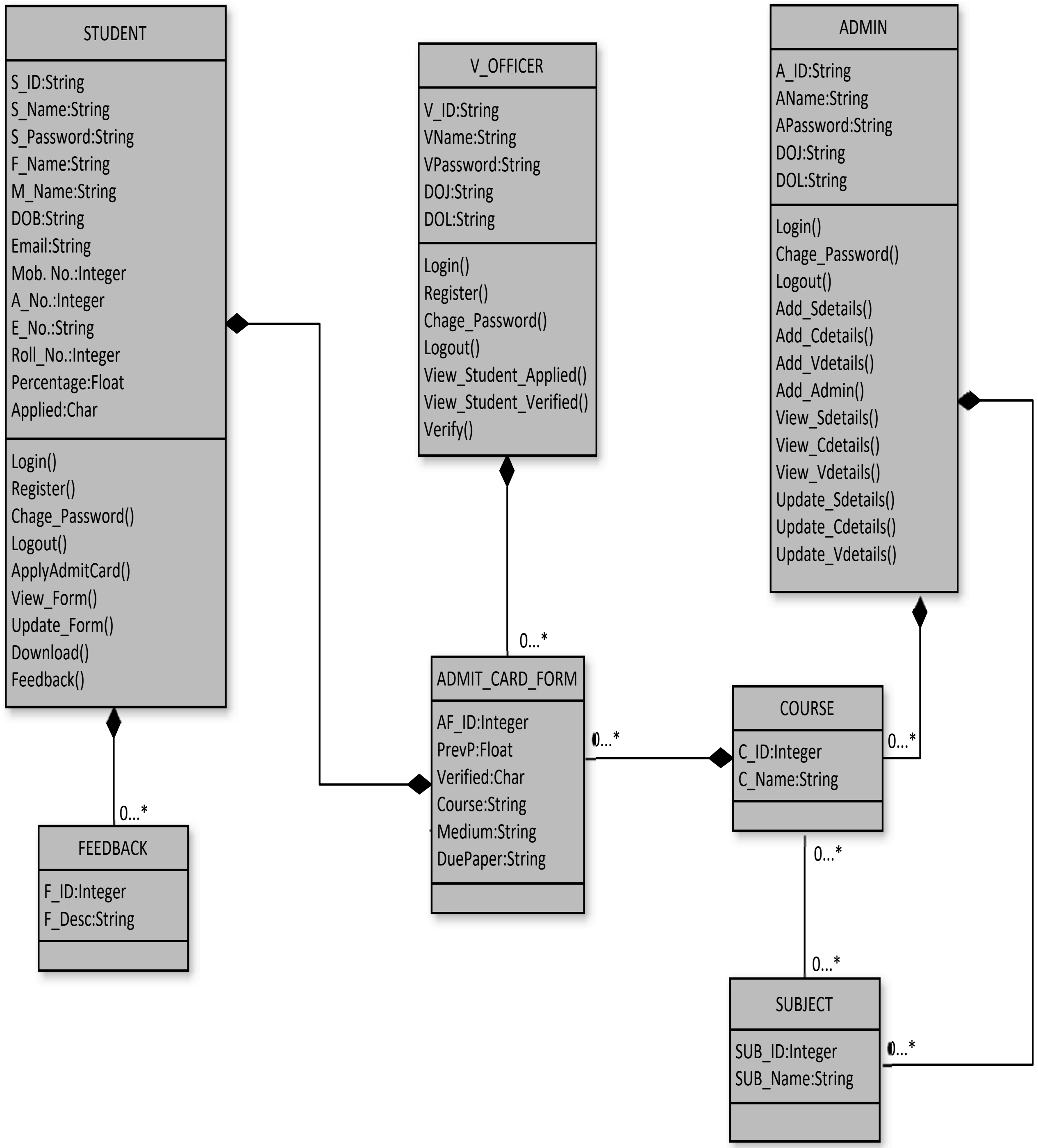
2. **Verification Officer**



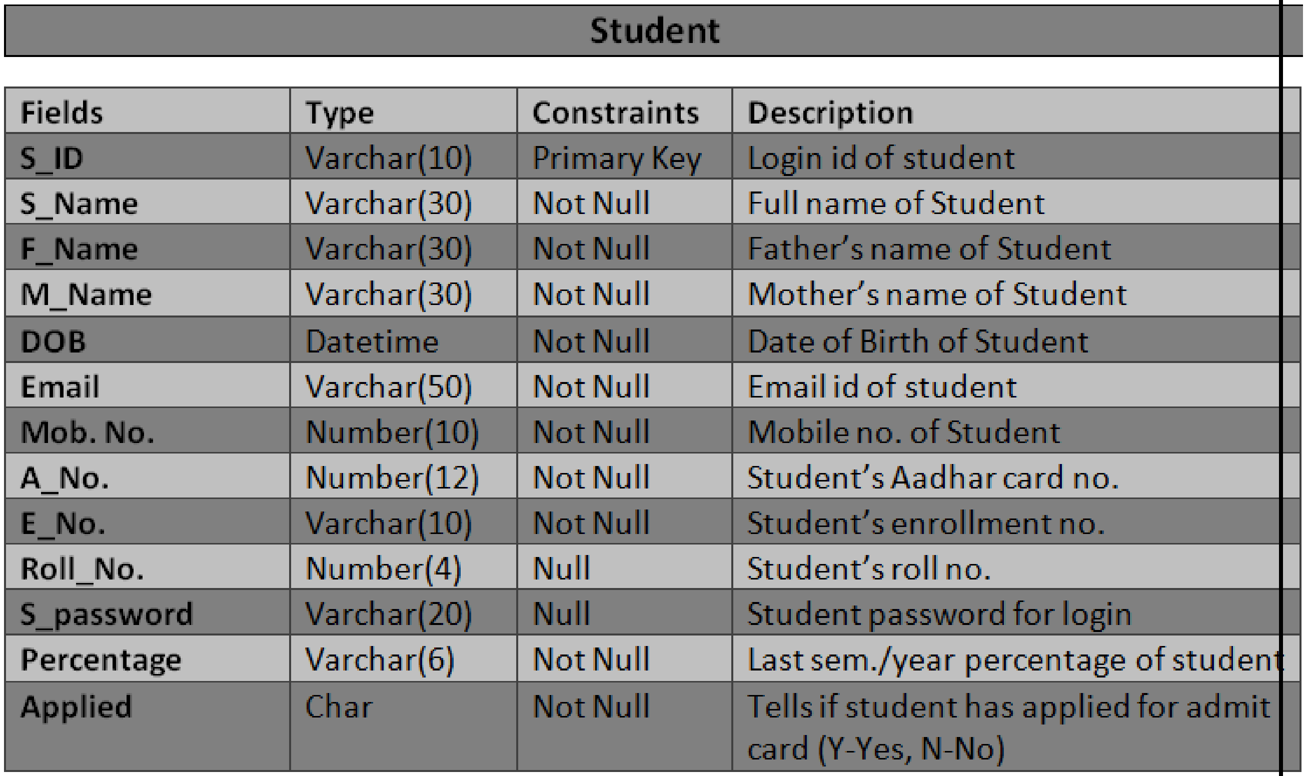
3.Admin



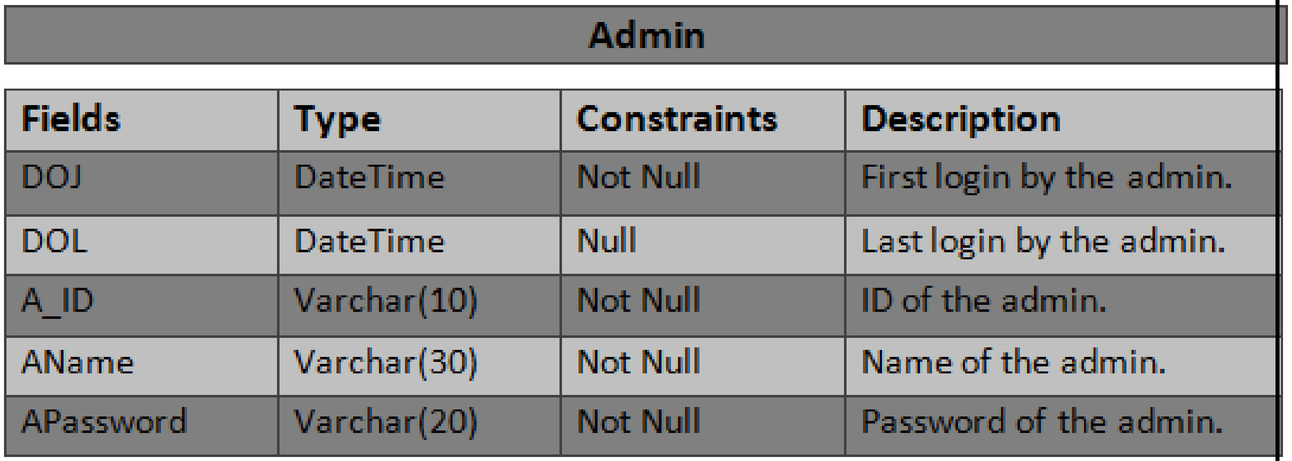
3.3 Class Diagram



3.4 Database Description



Student			
Fields	**Type**	**Constraints**	**Description**
S_ID	Varchar(10)	Primary Key	Login id of student
S_Name	Varchar(30)	Not Null	Full name of Student
F_Name	Varchar(30)	Not Null	Father's name of Student
M_Name	Varchar(30)	Not Null	Mother's name of Student
DOB	Datetime	Not Null	Date of Birth of Student
Email	Varchar(50)	Not Null	Email id of student
Mob. No.	Number(10)	Not Null	Mobile no. of Student
A_No.	Number(12)	Not Null	Student's Aadhar card no.
E_No.	Varchar(10)	Not Null	Student's enrollment no.
Roll_No.	Number(4)	Null	Student's roll no.
S_password	Varchar(20)	Null	Student password for login
Percentage	Varchar(6)	Not Null	Last sem./year percentage of student
Applied	Char	Not Null	Tells if student has applied for admit card (Y-Yes, N-No)



Admin			
Fields	**Type**	**Constraints**	**Description**
DOJ	DateTime	Not Null	First login by the admin.
DOL	DateTime	Null	Last login by the admin.
A_ID	Varchar(10)	Not Null	ID of the admin.
AName	Varchar(30)	Not Null	Name of the admin.
APassword	Varchar(20)	Not Null	Password of the admin.

V_Officer

Field	Type	Constraints	Description
DOJ	Datetime	Not Null	Joining date of verification officer.
DOL	Datetime	Null	Leaving date of verification officer.
V_ID	Varchar(10)	Null	ID of the verification officer.
VName	Varchar(20)	Not Null	Name of the verification officer.
VPassword	Varchar(10)	Null	Password of the verification officer.

Admit_card_form

Fields	Type	Constraints	Description
AF_ID	VarChar(4)	Not Null	Admit Card Form Number
PrevP	Number(3)	Not Null	Percentage of Pevious semester
Verified	Varchar(1)	Null	Depicts whether the admit card form is verified or not
Course	Varchar(30)	Not Null	Depicts the course in which student is enrolled.
S_ID	VarChar(10)	Not Null	ID of the Student
Medium	Varchar(10)	Not Null	Medium of examination
DuePaper	VarChar(1)	Not Null	Depicts if student has a due paper or not. (Y-Yes, N-No)

Feedback

Fields	Type	Constraints	Description
F_ID	Varchar(10)	Primary Key	Feedback no.
F_Desc	Varchar(50)	Not Null	Description of feedback
S_ID	Varchar(10)	Foreign Key	Student ID from which feedback has been submitted.

Course

Field	Type	Constraints	Description
C_ID	Varchar(10)	Primary Key	Course Id of all the courses.
C_Name	Varchar(20)	Not Null	Name of the course corresponding to its id.
AF_ID	Varchar(4)	Foreign Key	Admit Card Form Number
SUB_ID	Varchar(5)	Foreign Key	ID of the subject.
A_ID	Varchar(10)	Foreign Key	ID of the admin.

Subjects

Fields	Type	Constraints	Description
SUB_ID	Varchar(5)	Primary Key	Unique Id of the subject.
Sub_Name	Varchar(60)	Not Null	Name of the subject.
C_ID	Varchar(10)	Foreign Table	Unique Id of the course.

4.User Interface Diagrams (Coding)

Login Module-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
 <%@page import="java.sql.*"%>
<!DOCTYPE html>
<html>
    <head>
        <title>Sign In</title>
        <meta charset="UTF-8">
         <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
        <link rel="stylesheet" href="css/bootstrap-theme.min.css">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style>
        div#nbody2
        {background-color: rgba(255,255,255,0.5);
            width:770px;
            height: 100%;
            margin-left:auto;
            margin-right:auto;
            padding: 5px;
        }
        td
        {
            padding:0 10px -1px 7px;
        }
                 a {
    text-decoration: none;
    display: inline-block;
}
a:hover {
    color: white;
}
</style>
    </head>
    <body>
        <jsp:include page="header.jsp"></jsp:include>
        <br>
        <br><br>
        <div id="nbody2" style="text-align:center">
            <a href="login.jsp" style="float: left"><img src="back_arrow.jpg"  /></a>
            <p><h2><font face="calibri">ADMIN LOGIN</font></h2> </p>
            <center><form      name="form1"      method="post"      action="signinad.jsp"
class="form-group form-control-static form-inline    ">
                <table>
                    <tr>
                        <td><img src="login3.jpg" class="  img-rounded item "></td>
                        <td><input    type="text"    placeholder="            UserID"    name="u"
style="height:45px;width:250px" class="form-control   "></td>
                    </tr>
                    <tr>
                        <td>
                            <img src="login2.jpg" class="  img-rounded item " >
                        </td>
                        <td><input  type="password"  placeholder="       Password"  name="p"
style="height:45px;width:250px" class="form-control   "></td>
                    </tr>
                </table>
                      <%
                 String login_msg=(String)request.getAttribute("error");
                 if(login_msg!=null)
                out.println("<font color=red size=4px>"+login_msg+"</font>");
                  %><br>
          <input     type="submit"         value="LOGIN"     class="btn     btn-primary     "
/>&nbsp;&nbsp;

                    <input    type="reset"      value="RESET"    class="btn    btn-primary      "
/>&nbsp;&nbsp;

            </form></center>
        </div>
        </form>
    </body>
</html>

-----Coding-----------

<%@page contentType="text/html" pageEncoding="UTF-8"%>
    <%@page import="java.sql.*"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
    </body>
</html>
```

```
<%
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/admitdb" , "root" ,
"root");
        PreparedStatement                    ps=con.prepareStatement("SELECT
A_ID,A_Password,A_Name FROM admin WHERE A_ID=? AND A_Password=?
");
        ps.setString(1,request.getParameter("u"));
        ps.setString(2,request.getParameter("p"));
        ResultSet rs=ps.executeQuery();
        if(rs.next())
        {
            response.sendRedirect("admin_home.jsp");
            session.setAttribute("name",rs.getString(3));
            session.setAttribute("id",rs.getString(1));
            session.setAttribute("session","TRUE");
        }
        else
        {
             request.setAttribute("error","*Invalid UserID or Password");
            RequestDispatcher rd=request.getRequestDispatcher("/asign_in.jsp");
             rd.include(request, response);
            //response.sendRedirect("asign_in1.jsp");
        }
    }
    catch(Exception e){}
%>
```

Admit Form-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.sql.*,java.util.*"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Admit Card</title>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
        <link rel="stylesheet" href="css/bootstrap-theme.min.css">
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
        <script src="js/scripts.js"></script>
        <script src="js/jquery-1.11.3.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
        <script src="css/bootstrap.min.css"></script>
        <script>
            function showMyImage(fileInput) {
        var files = fileInput.files;
        for (var i = 0; i < files.length; i++) {
            var file = files[i];
            var imageType = /image.*/;
            if (!file.type.match(imageType)) {
                continue;
            }
            var img=document.getElementById("thumbnil");
            img.file = file;
            var reader = new FileReader();
            reader.onload = (function(aImg) {
                return function(e) {
                    aImg.src = e.target.result;
                };
            })(img);
            reader.readAsDataURL(file);
        }
    }
   $(document).ready(function () {
    //Initialize tooltips
    $('.nav-tabs > li a[title]').tooltip();
    //Wizard
    $('a[data-toggle="tab"]').on('show.bs.tab', function (e) {
        var $target = $(e.target);

        if ($target.parent().hasClass('disabled')) {
            return false;
        }
    });
    $(".next-step").click(function (e) {
        var $active = $('.wizard .nav-tabs li.active');
        $active.next().removeClass('disabled');
        nextTab($active);
    });
    $(".prev-step").click(function (e) {
        var $active = $('.wizard .nav-tabs li.active');
        prevTab($active);
    });
});
function nextTab(elem) {
    $(elem).next().find('a[data-toggle="tab"]').click();
}
```

```
function prevTab(elem) {
    $(elem).prev().find('a[data-toggle="tab"]').click();
}
        </script>
    </head>
    <jsp:include page="header.jsp"></jsp:include>
    <body>
        <br>
        <div class="container">
            <div class="row">
                                <section>
        <div class="wizard">
            <br><h1 style="text-align:center">Admit Card Form</h1>
            <div class="wizard-inner">
                <div class="connecting-line"></div>
                <ul class="nav nav-tabs" role="tablist">
                    <li role="presentation" class="active">
   <a href="#step1" data-toggle="tab" aria-controls="step1" role="tab" title="Step 1">
                            <span class="round-tab">
                                <i class="glyphicon glyphicon-folder-open"></i>
                            </span>
                        </a>
                    </li>

                    <li role="presentation" class="disabled">
   <a href="#step2" data-toggle="tab" aria-controls="step2" role="tab" title="Step 2">
                            <span class="round-tab">
                                <i class="glyphicon glyphicon-pencil"></i>
                            </span>
                        </a>
                    </li>
                    <li role="presentation" class="disabled">
   <a href="#step3" data-toggle="tab" aria-controls="step3" role="tab" title="Step 3">
                            <span class="round-tab">
                                <i class="glyphicon glyphicon-picture"></i>
                            </span>
                        </a>
                    </li>

                    <li role="presentation" class="disabled">
                        <a    href="#complete"    data-toggle="tab"    aria-controls="complete"
role="tab" title="Complete">
                            <span class="round-tab">
                                <i class="glyphicon glyphicon-ok"></i>
                            </span>
                        </a>
                    </li>
                </ul>
            </div>

            <form role="form">
                <div class="tab-content">
                        <div class="tab-pane active" role="tabpanel" id="step1">
                        <h3>&nbsp;&nbsp;Personal Details</h3>
                        <p>
                             <table class='table'>
        <%
        String id=(String)session.getAttribute("id");
        try
        {
          Class.forName("com.mysql.jdbc.Driver");
          Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/admitdb" , "root" , "root");
          String s=("select * from student where S_ID='"+id+"'");
          Statement stmt=conn.createStatement();
          Statement st=conn.createStatement();
          ResultSet rs1=stmt.executeQuery(s);
           while(rs1.next())
           {%>
           <col width="33%">
           <col width="33%">
           <tr>
               <td height='27'>
                   ROLL NO.:<input type='text' size="15" name='roll' value="<%=rs1.getString("Roll_No")%>" class="form-control" disabled>
               </td><td></td>
                <td height='27'>
                        ENROLLMENT NO.: <input type='text' name='enroll' value="<%=rs1.getString("Enroll_No")%>" class="form-control" disabled></div>
                </td>
           </tr>
           <tr>
                <td height='27'>
                        AADHAR NO.:<input type='text' name='aadhaar' class="form-control" value="<%=rs1.getString("Aadhar_No")%>" disabled>
                </td><td></td>
                <td>ID:<input type='text' name='id' class="form-control" disabled value="<%=rs1.getString("S_ID")%>"></td>
            </tr>
            <tr>
                <td></td>
                <td>NAME OF CANDIDATE:<input type='text' name='name'
class="form-control" disabled value="<%=rs1.getString("First_Name")%>
<%=rs1.getString("Mid_Name")%> <%=rs1.getString("Last_Name")%>"></td>
         </tr>
         <tr>
            <td></td>
            <td>FATHER'S NAME:<input type='text' name='fname' class="form-
control" disabled value="<%=rs1.getString("F_Name")%>"></td>
         </TR>
         <tr>
            <td></td>
            <td>MOTHERS'S NAME:<input type='text' name='mname' class="form-
control" disabled value="<%=rs1.getString("M_Name")%>"></td>
         </tr>
         <% } %>
         <tr>
            <td></td>
         <td>CLASS :<input type='text' name='cname' class="form-control" disabled
value="<%=request.getParameter("cl1")%>">
            <%
               ResultSet rs4;
               String cn=(String)request.getParameter("cl1");
                           rs4=st.executeQuery("select C_ID from course where
C_Name='"+cn+"'");
                           while(rs4.next()){ %>
                           <input name="cid" value="<%=rs4.getString("C_ID")%>"
type="hidden">
                  </td>
         </tr>
           </table>
                  </p>
                  <ul class="list-inline pull-right">
                     <li><button type="button" class="btn btn-primary next-step">Save
and continue</button></li>
                  </ul>
               </div>
               <div class="tab-pane" role="tabpanel" id="step2">
                  <h3>&nbsp;&nbsp;Educational Details</h3>
                  <p>
                     <table class="table">
               <col width="30%">
               <col width="40%">
            <tr>
              <td>
                 <table class="table">
                    <tr><td><b>Foundation Courses</b></td></tr>
                 <%
                    ResultSet rs5;
                    Statement stmt5=conn.createStatement();
                    Statement stmt4=conn.createStatement();
                           String c2=(String)request.getParameter("cl1");
                           rs4=stmt.executeQuery("select C_ID from course where
C_Name='"+c2+"'");
                           while(rs4.next()){
                              rs5=stmt4.executeQuery("select SUB_ID from
course_subject where C_ID='"+rs4.getString("C_ID")+"' ");
                                while(rs5.next())
                           {
                              rs1=stmt5.executeQuery("select * from subject where
SUB_ID='"+(rs5.getString("SUB_ID") )+"' and Type='F' ");
                              while(rs1.next())
                              { %>
                                <tr><td><input type='text' name='dname' class="form-
control" disabled value="<%=rs1.getString("SUB_NAME")%>"></td> </tr>
         <% } } }%>
                         <tr><td><b>Five Fold Activities</b></td></tr>
                         <tr>
                           <td height='30'>
                              Subject1:<input type='text' name='ff1' class="form-control">
                              </td>
                         </tr>
                         <tr>
                              <td height='30'>
                              Subject2:<input type='text' name='ff2' class="form-control">
                              </td>
                         </tr>
                         <tr>
                              <td height='30'>
                              Subject3:<input type='text' name='ff3' class="form-control">
                              </td>
                         </tr>
                 </table>
              </td>
              <td>
                 <table class="table">
                      <tr><td><b>Disciplinary Courses</b></td></tr>
               <%
                           String c=(String)request.getParameter("cl1");
                           rs4=stmt.executeQuery("select C_ID from course where
C_Name='"+c+"'");
                           while(rs4.next()){
```

```
                              rs5=stmt4.executeQuery("select SUB_ID from course_subject where C_ID='"+rs4.getString("C_ID")+"' ");
                             while(rs5.next())
                            {
                              rs1=stmt5.executeQuery("select * from subject where SUB_ID='"+(rs5.getString("SUB_ID") )+"' and Type='D' ");
                              while(rs1.next())
                              { %>
              <tr><td><input type='text' name='dname' class="form-control" disabled value="<%=rs1.getString("SUB_NAME")%>"></td> </tr>
               <% }
}   }%>
               </table>
               </td>
               <td>
                  <table class="table">
                     <tr><td><b>Practical Subjects</b></td></tr>
                       <%
                            String c1=(String)request.getParameter("cl1");
                            rs4=stmt.executeQuery("select C_ID from course where C_Name='"+c1+"'");
                            while(rs4.next()){
                              rs5=stmt4.executeQuery("select SUB_ID from course_subject where C_ID='"+rs4.getString("C_ID")+"' ");
                             while(rs5.next())
                            {
                              rs1=stmt5.executeQuery("select * from subject where SUB_ID='"+(rs5.getString("SUB_ID") )+"' and Type='P' ");
                              while(rs1.next())
                              { %>
              <tr><td> <input type='text' name='psub' class="form-control" disabled value="<%=rs1.getString("SUB_NAME")%>"> </td></tr>
         <% } } }%>
                  </table>
               </td>
            </tr>
           <%
              String s1=("select * from student where S_ID='"+id+"'");
               rs1=stmt.executeQuery(s1);
        while(rs1.next())
        {%>
           <tr><td></td>
               <td>PREVIOUS SEMESTER %:<input type='text' name='percent' class="form-control" disabled value="<%=rs1.getString("Percentage")%>"/></td>
           </tr>
           <tr><td></td>
```

```
          <td>MEDIUM:<input type='text' name='medium' class="form-control">
</td>
        </tr>
        <tr><td></td>
          <td>DUE PAPER:<input type='text' name='due' class="form-control">
</td>
        </tr>
        <% }%>
        </table>
             </p>
             <ul class="list-inline pull-right">
               <li><button type="button" class="btn btn-default prev-
step">Previous</button></li>
               <li><button type="button" class="btn btn-primary next-step">Save
and continue</button></li>
             </ul>
           </div>

           <div class="tab-pane" role="tabpanel" id="step3">
             <h3>&nbsp;&nbsp;Declaration</h3>
             <p>&nbsp;&nbsp;&nbsp;&nbsp; I hereby declare that the details
furnished above are true and correct to the best of my knowledge
and belief and I undertake to inform you of any changes therein, immediately. In case
any of the above
information is found to be false or untrue or misleading or misrepresenting, I am
aware that I may be held liable
for it.
             <table class='table' align='right'>
               <col width='60%'>
               <tr><td>
                   <table class='table table-borderless'>
                     <col width='10%'>
               <col width="40%">
                     <tr><td>Place:</td><td><input NAME="place"
type="text"></td></tr>
                     <tr><td>Date:</td><td><input NAME="date"
type="text"></td></tr>
                     <tr>
                       <%
                         rs1=stmt.executeQuery(s);
                         while(rs1.next())
                          {%>
                          <td>Signature:</td><td><input NAME="date"
type="text" value=<%=rs1.getString("First_Name")%> disabled></td>
                          <% } %>
                     </tr>
```

```
                                </table></td>
                            <td height='27'>
                    PHOTOGRAPH:<input NAME="file" type="file" accept="image/*"
class="form-control" onchange="showMyImage(this)">
                    <img id="thumbnil" style="width:20%; margin-top:10px;" src=""
alt="image"/>
                </td></tr>
                        </table>
                        </p>
                        <ul class="list-inline pull-right">
                            <li><button type="button" class="btn btn-default prev-
step">Previous</button></li>
                                  <li><button type="button" class="btn btn-success btn-info-full
next-step">Submit</button></li>
                        </ul>
                    </div>
                    <div class="tab-pane" role="tabpanel" id="complete">
                        <h3>&nbsp;&nbsp;Complete</h3>
                        <p><font size='5px' color='green'><center>You have successfully
completed all steps.</center></font></p>
                        <ul class="list-inline pull-right">
                            <li><button type="submit" class="btn btn-info btn-info-full"
formaction="admitSubmit.jsp">OK</button></li>
                        </ul>
                    </div>
                    <div class="clearfix"></div>
                </div>

                     <%
                           }
            catch(Exception e){
               response.sendRedirect("error.jsp");
              }
            %>
             </form>
        </div>
    </section>
   </div>
</div>
        </div>
    </body>
</html>
```

---Coding----

```
<%@page import="java.sql.*,java.util.*"%>
```

```
<%@page import="java.io.FileOutputStream"%>
<%@page import="java.io.File"%>
<%@page import="java.io.DataInputStream"%>
<%@page import="java.io.InputStream"%>
<%@page import="java.io.FileInputStream"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%! static int af_id=0; %>
    <%
        try
        {
          Class.forName("com.mysql.jdbc.Driver");
          Connection                                    conn                                    =
DriverManager.getConnection("jdbc:mysql://localhost:3306/admitdb"    ,    "root"    ,
"root");
          String class1=request.getParameter("cid");
          String med=request.getParameter("medium");
          String due=request.getParameter("due");
          String id=(String)session.getAttribute("id");
          String sub1=request.getParameter("ff1");
          String sub2=request.getParameter("ff2");
          String sub3=request.getParameter("ff3");
          String p=request.getParameter("place");
          String d=request.getParameter("date");
          String file1=request.getParameter("file");
          String f2="C:/Users/Deepti Dixit/Pictures/docs/img.jpg";
           Statement st = conn.createStatement();
           /*String sql = "update student set Pic=?, where S_ID='"+id+"' ";
           out.println(file1);
            PreparedStatement statement = conn.prepareStatement(sql);
            out.println("4");
            InputStream  inputStream  =  new  FileInputStream(new  File("C:/Users/Deepti
Dixit/Pictures/docs/img.jpg"));
            out.println("1");
            statement.setBlob(1,inputStream);
            out.println("2");
            int row = statement.executeUpdate();
            out.println("3");*/
           ResultSet rs=st.executeQuery("select MAX(AF_ID) from admit_card ");
          while(rs.next())
```

```
            af_id=rs.getInt(1);
          ++af_id;
         int i=st.executeUpdate("insert into admit_card(C_ID,Medium,Due_Paper,S_ID,FiveFold1,FiveFold2,FiveFold3,Place,AF_ID,Date)"
            + " values ('"+class1+"','"+med+"','"+due+"','"+id+"','"+sub1+"','"+sub2+"','"+sub3+"','"+p+"','"+af_id+"','"+d+"')");
          int j=st.executeUpdate("update student set Applied='Y' where S_ID='"+id+"'");
          response.sendRedirect("student_home.jsp");
        }
        catch(Exception e)
        {
          //response.sendRedirect("error.jsp");
        }
    %>
    </body>
</html>
```

View V_Officer:

```
<%@page import="javax.jms.Session"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<!DOCTYPE html>
<html>
    <head>
        <title>V_Officer Details</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
        <link rel="stylesheet" href="css/bootstrap-theme.min.css">
        <style>
        div#nbody3
        {background-color: rgba(255,255,255,0.5);
            width:770px;
            height: 100%;
            margin-left:auto;
            margin-right:auto;
            padding: 5px;
        }
        div#n3
        {
            width:470px;
            height: 100%;
        }
        td
        {
            padding:0 10px -1px 7px;
        }
.dropbtn {
         background: url("propic2.png");
         height: 80px;
         width: 80px;
          color: white;
           padding: 16px;
            font-size: 16px;
         cursor: pointer;
}
/* Dropdown button on hover & focus */
.dropbtn:hover, .dropbtn:focus {
    background: url("propic2.png");
    height: 80px;
    width: 80px;
}
/* The container <div> - needed to position the dropdown content */
.dropdown {
    position: relative;
    display: inline-block;
}
/* Dropdown Content (Hidden by Default) */
.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f1f1f1;
    min-width: 160px;
    box-shadow: 0px 10px 15px 0px rgba(0,0,0,0.2);
    z-index: 1;
}
/* Links inside the dropdown */
.dropdown-content a {
    color: blue;
    padding: 10px 10px;
    text-decoration: buttontext;
    display: block;
}
/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color: #fdd}
/* Show the dropdown menu (use JS to add this class to the .dropdown-content
container when the user clicks on the dropdown button) */
```

```
.show {display:block;}
a {
    text-decoration: none;
    display: inline-block;

}
a:hover {

    color: white;
}
</style>
<script src="js/jquery-1.11.3.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
<script>
    function myFunction() {
    document.getElementById("myDropdown").classList.toggle("show");
}
// Close the dropdown menu if the user clicks outside of it
window.onclick = function(event) {
  if (!event.target.matches('.dropbtn')) {
    var dropdowns = document.getElementsByClassName("dropdown-content");
    var i;
    for (i = 0; i < dropdowns.length; i++) {
      var openDropdown = dropdowns[i];
      if (openDropdown.classList.contains('show')) {
        openDropdown.classList.remove('show');
      }  }
  }
}
function go(){

window.location.replace("logout.jsp",'window','toolbar=1,location=1,directories=1,st
atus=1,menubar=1,scrollbars=1,resizable=1');
    self.close();
}
</script>
    </head>
    <body>
        <jsp:include page="header.jsp"></jsp:include>
        <br>
        <div id="nbody3">
            <a href="admin_home.jsp" style="float: left" title="Back"><img
src="back_arrow.jpg"  /></a>
        <table>
                <tr>
                    <td width='120%'></td>
                <td>
                    <h2><button onclick="myFunction()" class="dropbtn" ></button>
            <div id="myDropdown" class="dropdown-content">
                <font size="2"><a href="schangePassword.jsp">Change Password</a>
                <a href="javascript:go()">Logout</a></font>
            </div>
                </h2>
                    </td>
                </tr>
            </table>
        <br>
         <center>
            <%
                String id= null;
            try{
           Class.forName("com.mysql.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/admitdb" , "root" ,
"root");
            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery("select * from v_officer");
        %>
        <center>
            <div id='n3'>
            <table class="table table-bordered table-striped">
            <tr>
                <th width='100px'>V_ID</th>
                <th width='200px'>V_Name</th>
                <th width='20px'></th>
                <th width='20px'></th>
            </tr>
            <%
              while(rs.next()){
             %>
             <tr>
                 <td><%=rs.getString(1)%></td>
                 <td><%=rs.getString(2)%></td>
                 <td><a      href="del_v.jsp?uid=<%=rs.getString(1)%>"      onclick="return
confirm('Are you sure you want to delete?')"><img src="del1.png"</a></td>
                 <td><a
href="edit_v.jsp?uid=<%=rs.getString(1)%>&uname=<%=rs.getString(2)%>&pid=<
%=rs.getString(1)%>"><img src="edit1.png"></a></td>
             </tr>
             <% }
             %>
            </table><br><br>
        <%
                rs.close();
                st.close();
                con.close();
                }
                catch(Exception e){
                      response.sendRedirect("error.jsp");
                }
                %>
                <form method="get" action="add_v.jsp">
                    <button type="submit" class="btn btn-primary"
style="height:30px;width:150px">Add</button>
   </form>
            </div>
   </center>
        </div>
   </body>
</html>
```

Add V_Oficer:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
   </head>
   <body>

<%
String connectionURL = "jdbc:mysql://localhost:3306/admitdb";
 try {
     String d=(String)request.getParameter("vi");
     String p=(String)request.getParameter("vn");
     String pa=(String)request.getParameter("vp");
     Class.forName("com.mysql.jdbc.Driver");
 Connection conn = DriverManager.getConnection(connectionURL, "root", "root");
     Statement st = conn.createStatement();
      int i=st.executeUpdate("insert into v_officer (V_ID,V_Name,V_Password) values
('"+d+"','"+ p+"','"+ pa+"')");
      response.sendRedirect("view_vofficer.jsp");
       conn.close();
 }
 catch (SQLException ex) {
     response.sendRedirect("error.jsp");
 }
%>
    </body>
</html>
```

Update V_Officer-

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
<%
String connectionURL = "jdbc:mysql://localhost:3306/admitdb";
 try {
     String pd=(String)request.getParameter("pid");
     String d=(String)request.getParameter("vi");
     String p=(String)request.getParameter("vn");
     String pa=(String)request.getParameter("vp");
     Class.forName("com.mysql.jdbc.Driver");
 Connection conn = DriverManager.getConnection(connectionURL, "root", "root");
     Statement st = conn.createStatement();
      int  i=st.executeUpdate("update  v_officer  set  V_ID='"+d+  "',V_Name='"+p+  "'
where V_ID='"+pd+"'");
     response.sendRedirect("view_vofficer.jsp");
       conn.close();
 }
 catch (SQLException ex) {
     response.sendRedirect("error.jsp");
 }
%>
    </body>
</html>
```

Delete V_Officer-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.sql.*"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
<%
String connectionURL = "jdbc:mysql://localhost:3306/admitdb";
 try
 {
     String uid=request.getParameter("uid");
     Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection(connectionURL, "root", "root");
     Statement st = conn.createStatement();
     st.executeUpdate("delete from v_officer where V_ID='"+uid+"' ");
     response.sendRedirect("view_vofficer.jsp");
       conn.close();
 }
 catch (SQLException ex) {
     response.sendRedirect("error.jsp");
 }
%>
    </body>
</html>
```

5. Testing Issues

Software testing can be stated as the process of validating ad verifying that a software program/ application/ product:

1. Meets the requirements that guided its design and development;
2. Works as expected; and
3. Can be implemented with the same characteristics.

Testing is the process of making sure that the program performs the intended tasks.

- Unit Testing: Each component or part of the system is tested individually.

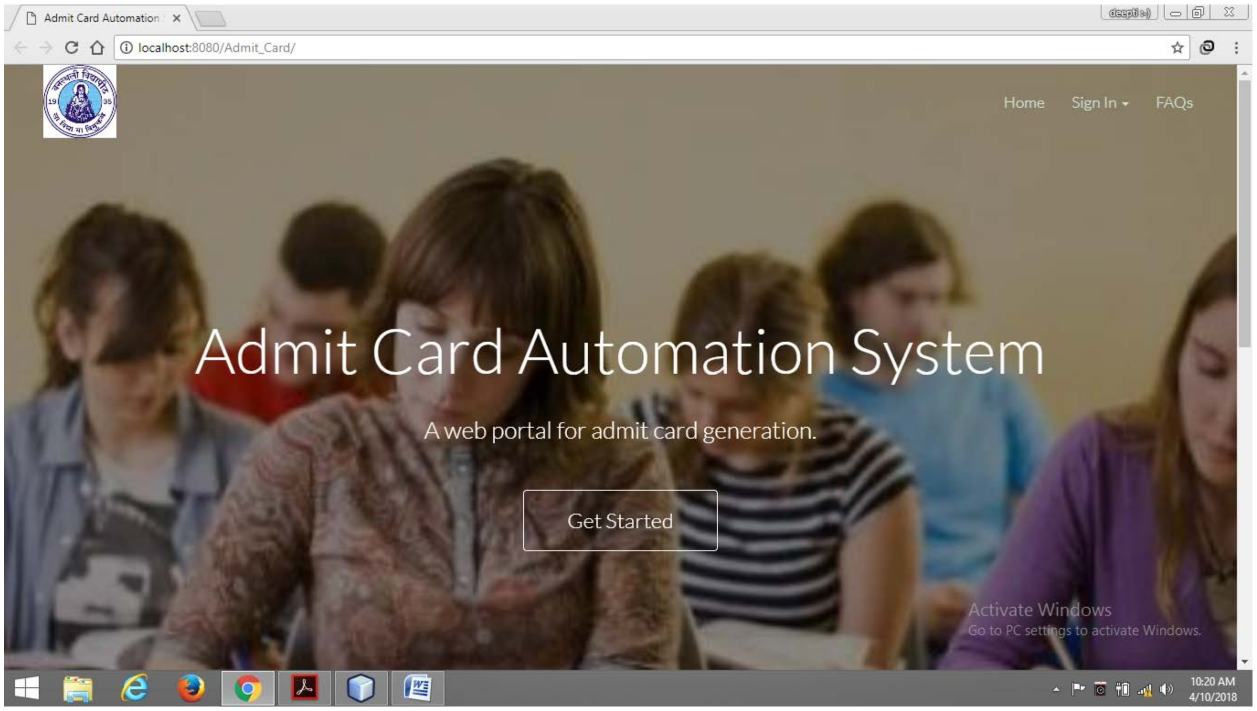
- Module Testing: A collection of dependent components such as an object class, procedures and functions are tested in this testing.

- Integration/Regression Testing: In this, many units tested modules are combined into sub systems, which are then tested.

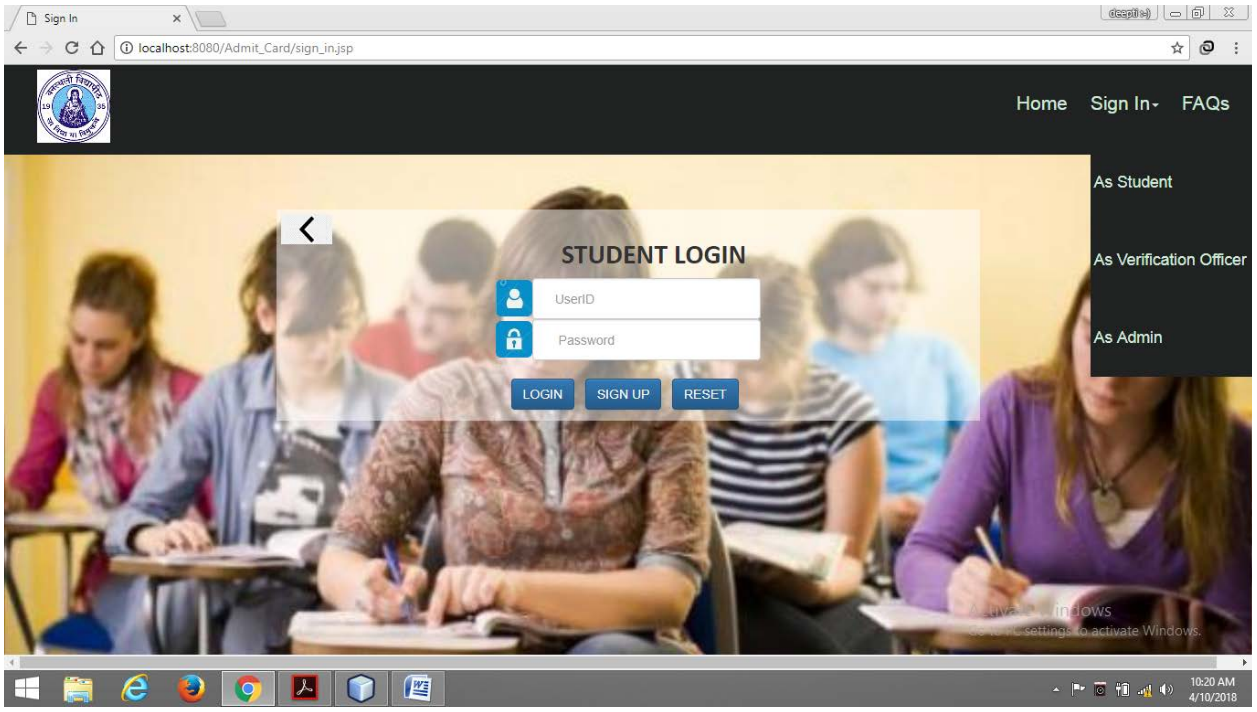
- System Testing: Entire system software is tested. It is a testing of the system against its initial objectives.

6.User Interfaces

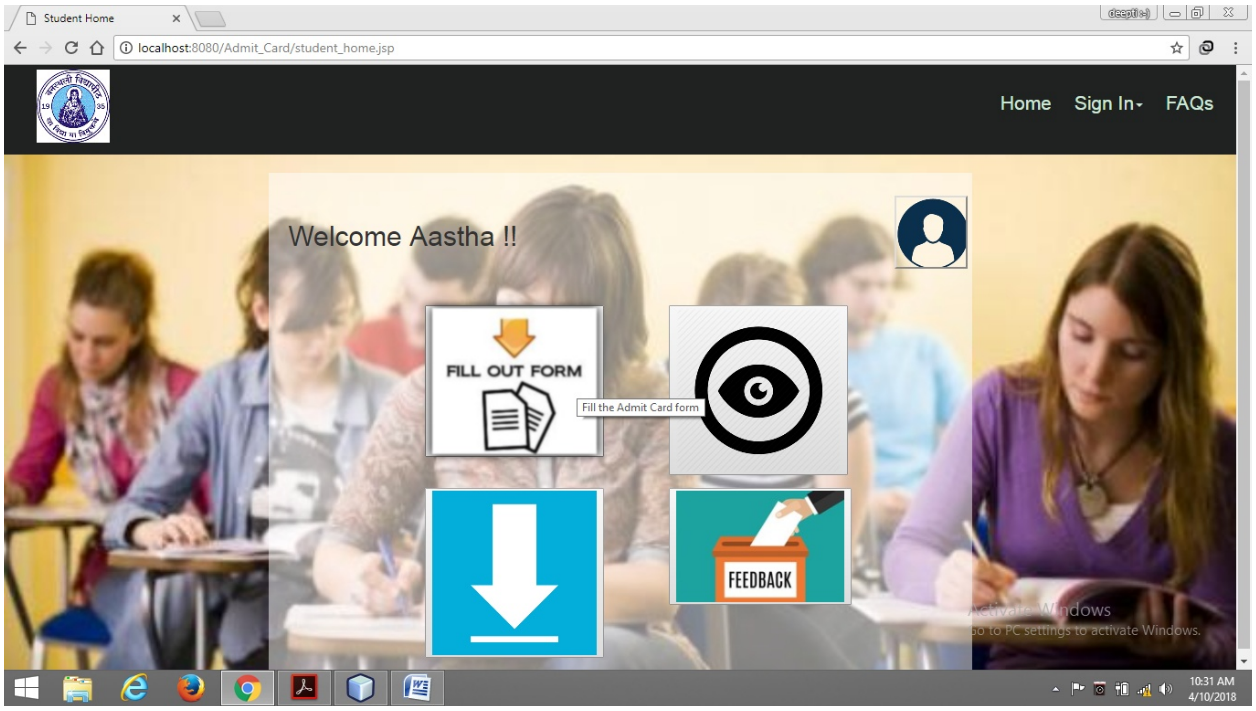
1.Home Page:


Admit Card Automation
localhost:8080/Admit_Card/
Home
Sign In
FAQs
Admit Card Automation System
A web portal for admit card generation.
Get Started
Activate Windows
Go to PC settings to activate Windows.
10:20 AM
4/10/2018

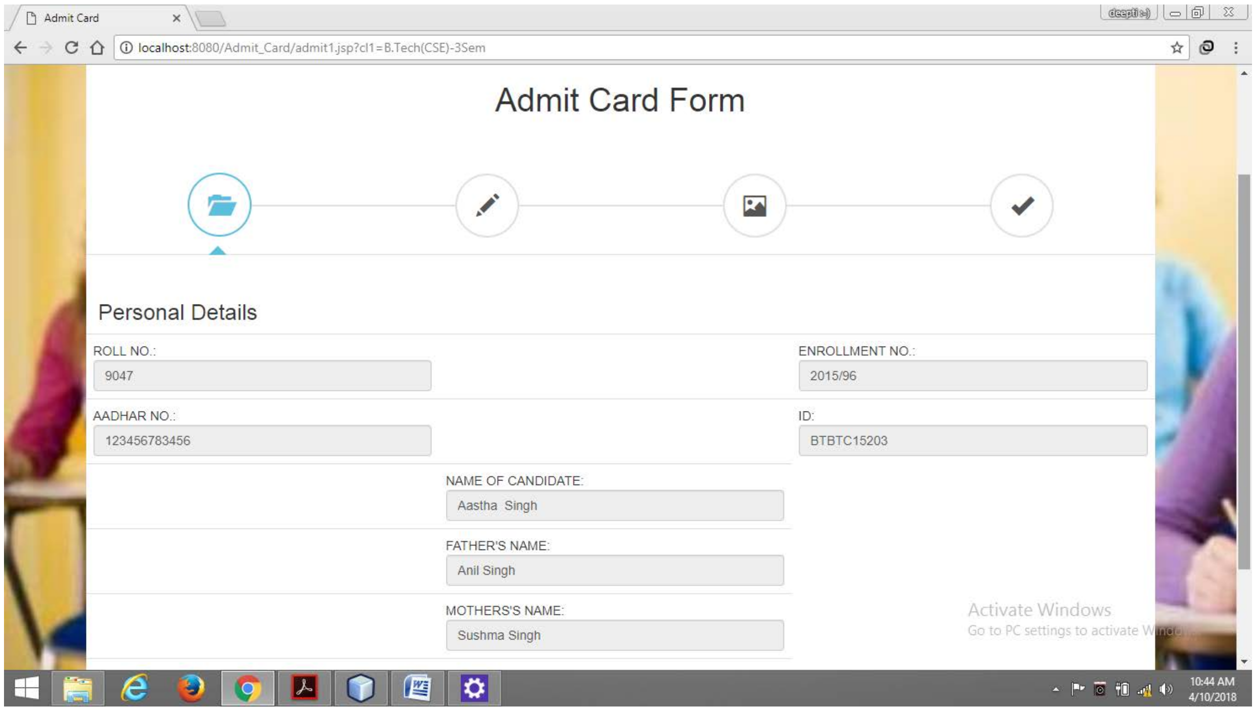
2.Sign In:



3.Student Home:



4.Admit Card Form Fill:



Admit Card Form

Personal Details

ROLL NO.: 9047

ENROLLMENT NO.: 2015/96

AADHAR NO.: 123456783456

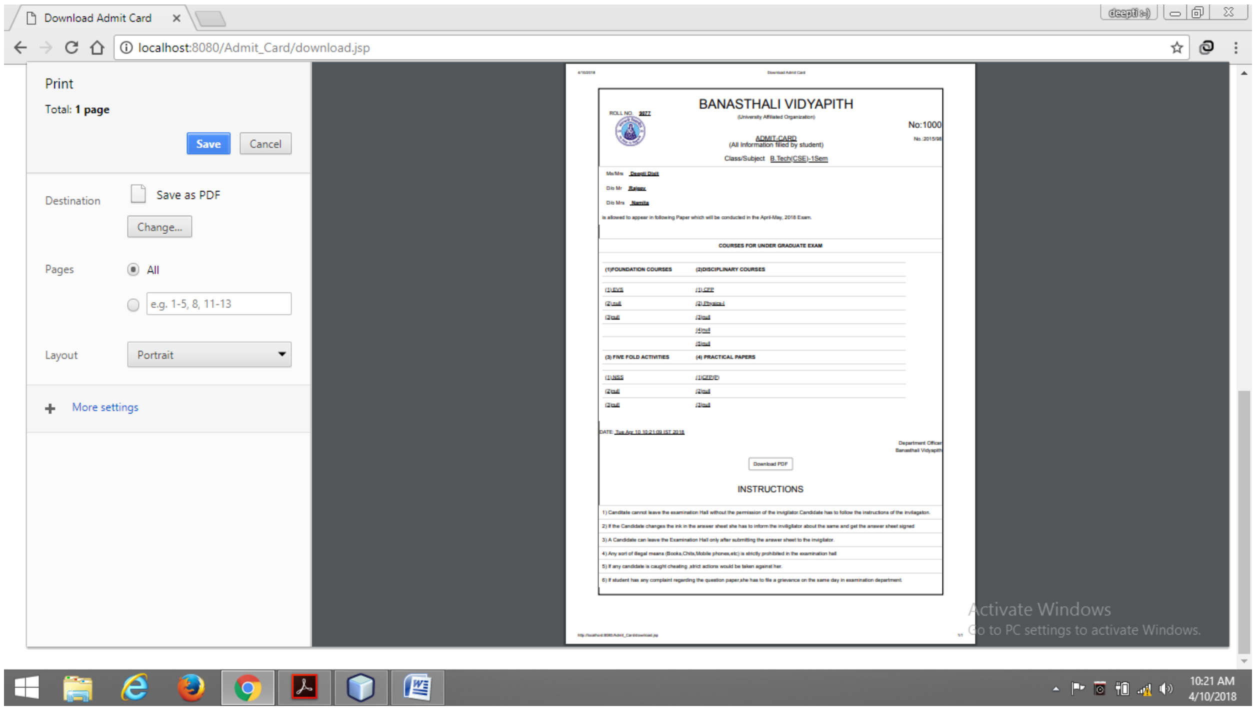
ID: BTBTC15203

NAME OF CANDIDATE: Aastha Singh

FATHER'S NAME: Anil Singh

MOTHERS'S NAME: Sushma Singh

5.Download Admit Card:



localhost:8080/Admit_Card/download.jsp

Print
Total: **1 page**
Destination: Save as PDF
Pages: All
Layout: Portrait

BANASTHALI VIDYAPITH
(University Affiliated Organization)

ROLL NO. 9077

No:1000
No.:2015/98

ADMIT-CARD
(All Information filled by student)

Class/Subject B.Tech(CSE)-1Sem

Ms/Mrs Deepti Dixit
D/o Mr Rajeev
D/o Mrs Namita

is allowed to appear in following Paper which will be conducted in the April-May, 2018 Exam.

COURSES FOR UNDER GRADUATE EXAM

(1)FOUNDATION COURSES	(2)DISCIPLINARY COURSES
(1) EVS	(1) CFP
(2) null	(2) Physics-I
(3)null	(3)null
	(4)null
	(5)null
(3) FIVE FOLD ACTIVITIES	**(4) PRACTICAL PAPERS**
(1) NSS	(1)CFP(P)
(2)null	(2)null
(3)null	(3)null

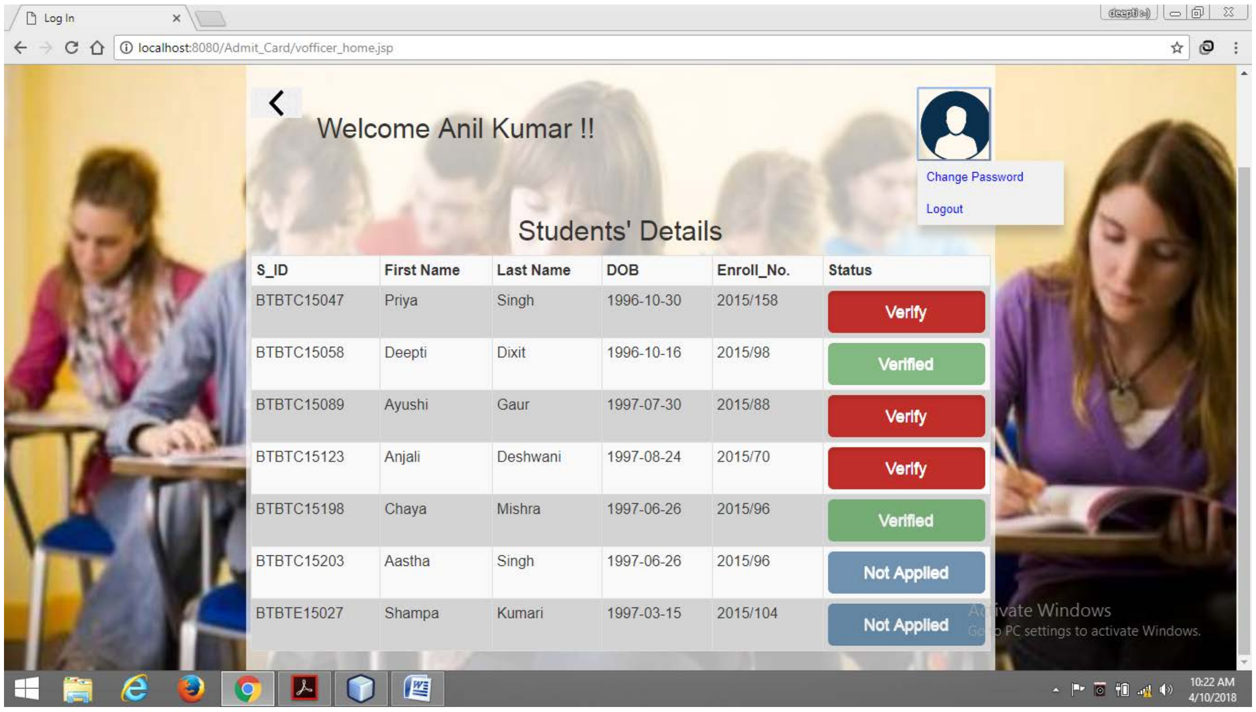
DATE: Tue Apr 10 10:21:09 IST 2018

Department Officer
Banasthali Vidyapith

INSTRUCTIONS

1) Candidate cannot leave the examination Hall without the permission of the invigilator.Candidate has to follow the instructions of the invilagator.
2) If the Candidate changes the ink in the answer sheet she has to inform the invilligilator about the same and get the answer sheet signed
3) A Candidate can leave the Examination Hall only after submitting the answer sheet to the invigilator.
4) Any sort of illegal means (Books,Chits,Mobile phones,etc) is strictly prohibited in the examination hall
5) If any candidate is caught cheating ,strict actions would be taken against her.
6) If student has any complaint regarding the question paper,she has to file a grievance on the same day in examination department.

6. V_Officer Home:



localhost:8080/Admit_Card/vofficer_home.jsp

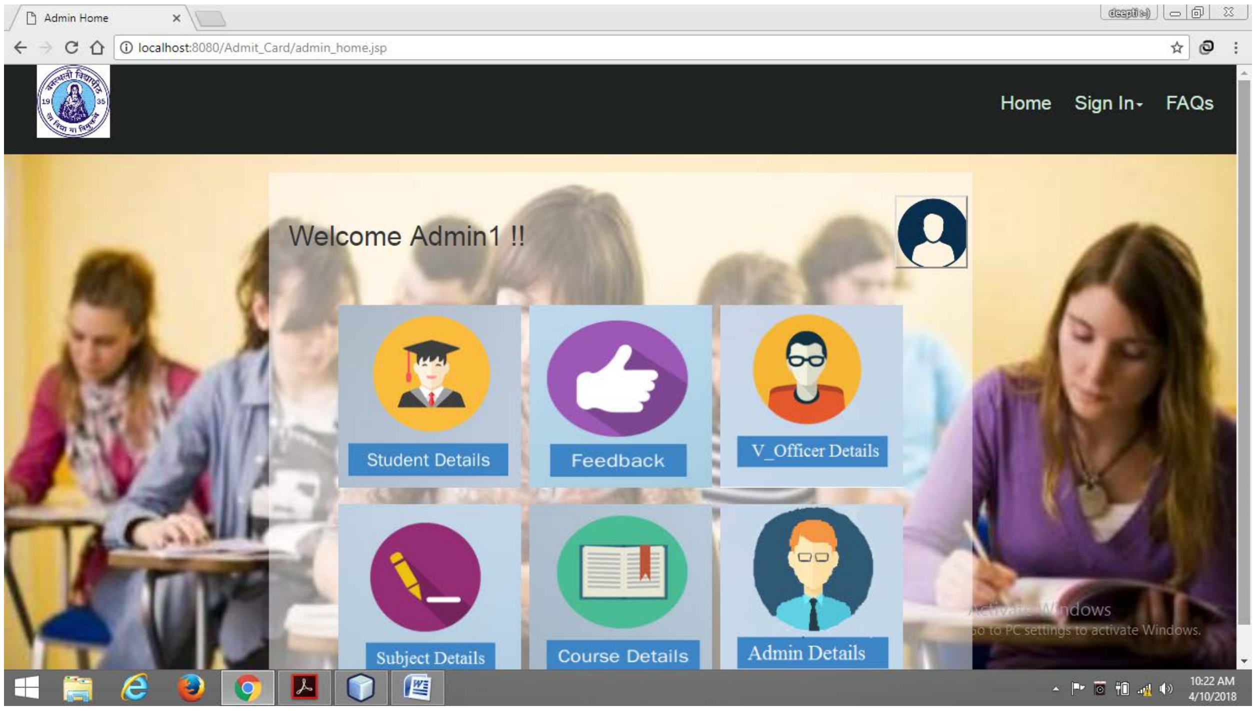
Welcome Anil Kumar !!

Change Password
Logout

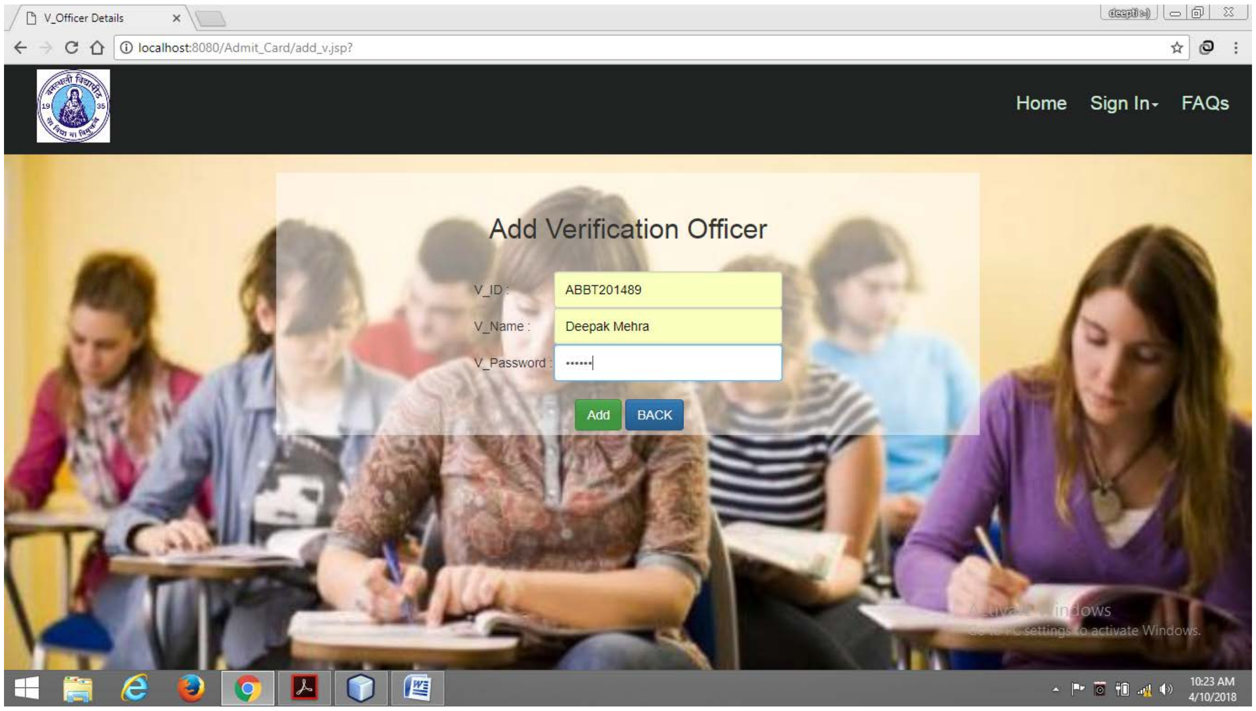
Students' Details

S_ID	First Name	Last Name	DOB	Enroll_No.	Status
BTBTC15047	Priya	Singh	1996-10-30	2015/158	Verify
BTBTC15058	Deepti	Dixit	1996-10-16	2015/98	Verified
BTBTC15089	Ayushi	Gaur	1997-07-30	2015/88	Verify
BTBTC15123	Anjali	Deshwani	1997-08-24	2015/70	Verify
BTBTC15198	Chaya	Mishra	1997-06-26	2015/96	Verified
BTBTC15203	Aastha	Singh	1997-06-26	2015/96	Not Applied
BTBTE15027	Shampa	Kumari	1997-03-15	2015/104	Not Applied

7.Admin Home:



8.Add Module:



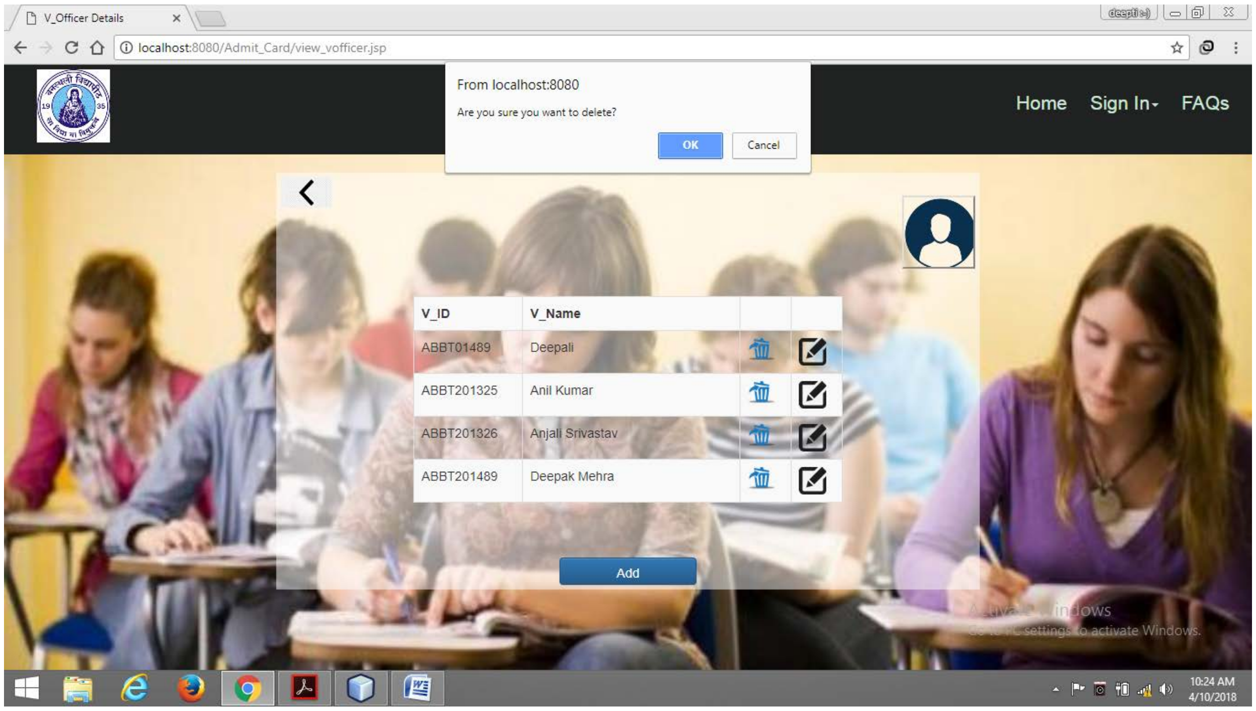
Add Verification Officer

V_ID : ABBT201489

V_Name : Deepak Mehra

V_Password : ••••••

9.View/Delete Module:



From localhost:8080

Are you sure you want to delete?

V_ID	V_Name		
ABBT01489	Deepali		
ABBT201325	Anil Kumar		
ABBT201326	Anjali Srivastav		
ABBT201489	Deepak Mehra		

7.APPENDICES

ADMIN: Administrator is the controller of the entire web application who maintains the all records of the functionalities and the users.

GUEST USER: People who can browse only the Home Page and give feedback of the web application.Here, we have 2 categories of guest user which will remain same throughout. They are:

i. **Verification Officer**
ii. **Student**

REGISTERED USER

A user who has signed up for an account and has logged in to the web application. These users can also drop their feedbacks (if any).

There are 2 categories of registered users based on their differential privileges:

i. **TYPE-I(Admin)**

   Type-I can enrolled new student,view and update student details. He/She will also maintain web application modules.He/She will also update the course details.

ii. **TYPE-II(Verification Officer)**

   Type-II can verify the detail entered by the student in admit card form and also generate the admit card for the registered student.

iii. **TYPE-III(Student)**

   Type-III can apply for the admit card and also can download the admit card.

IEEE:

Institute of Electrical and Electronics Engineers.

Server:

The main computer on Network.

Browser:

A Software application used to locate and display the web pages.

SRS:

Software Requirement Specification.

HTML:

Hyper Text Markup Language is a markup language used to design static web pages.

WWW

World Wide Web is a network of online content that is formatted in HTML and accessed via HTTP. The term refers to all the interlinked HTML pages that can be accessed over the Internet.

HTTP

Hyper Text Transfer Protocol is the underlying protocol used by the World Wide Web to define how messages are formatted and transmitted.

JAVA

Java is a programming language that developers use to create applications on computers.

CSS

Cascading Style sheets are used to format the layout of web pages. They can be used to define text styles, table sizes and other aspects of web pages and helps web developers create a uniform look across several pages of a web site.

JSP

Java Server Pageis a server-side scripting language that enables the development of dynamic web sites.

TCP/IP

Transmission Control Protocol/Internet Protocol, the suite of communication protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being TCP and IP.

IE

Internet Explorer

RAM

Random Access Memory

DBA

Database Administrator

HDD

Hard Disk Drive

APACHE TOMCAT

Apache Tomcat is an application server meant to serve java applications.

8.REFERENCES

- IEEE Recommended Practice for Software Requirements Specification- IEEE Std. 830-1993.
- Pressman Roger S., Software Engineering "A Practitioner's Approach" Fifth Edition, McGraw-Hill Publication, 2000.
- Navathe Shamkant B., Fundamentals of Database Systems, Fifth Edition, Pearson Publication.
- Bayross Ivan, SQL, PL/SQL ,Third Edition, BPB Publication.
- IEEE STD 830-1998, IEEE Recommended Practice for Software Requirement Specifications.
- James Rumbaugh, Grady Booch -The Unified Modeling Language ,Second Edition.