

COLLEGE ADMIN MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted By

NITIKA PATEL

(University Roll No: 2000290140078)

ABHINAV SHARMA

(University Roll No: 2000290140005)

**Submitted in partial fulfillment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of

Mrs. Vidhushi. Mishra

ASSITANT PROFRSSOR

KITE Group of Institution, Ghaziabad



Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS

KIET Group of Institutions, Ghaziabad

Uttar Pradesh-201206

(MAY 2022)

CONTENTS

Chapter 1 - Introduction

- 1.1 Project description
- 1.2 Project Scope
- 1.3 Hardware / Software used in Project

Chapter 2 Feasibility Study

- 2.1 Technical feasibility
- 2.2 Operational Feasibility
- 2.3 Behavioral Feasibility
- 2.4 Operational Feasibility

Chapter 3 Database Design

- 3.1 Database Tables
- 3.2 Flow Chart
- 3.3 Use Case Diagram
- 3.4 Sequence Diagram
- 3.5 Collaborative Diagram
- 3.6 Entity Relationship Diagram
- 3.7 Data Flow Diagram

Chapter 4 Form Design

- 4.1 Input / Output Form (Screenshot)

Chapter 5 Coding

- Module wise code

Chapter 6 Testing

- 6.1 Test Case-1
- 6.2 Test Case-2

Bibliography

CANDIDATE DECLARATION

We hereby declare that the work, which is being presented in the project, entitled **College Admin Management System** towards the partial fulfillment of the requirement for the award of the degree of **MASTER of COMPUTER APPLICATION** in the department of **KIET Group of Institution, Ghaziabad (India)** is an authentic record of my own work carried out during the period from October 2021 to December 2021, under the guidance of **Mrs. Vidhushi .Mishra, Assistant Professor KIET Group of Institution, Ghaziabad** I have not submitted the matter embodied in this project for the award of any other degree or diploma.

NITIKA PATEL (2000290140078)
ABHINAV SHARMA (2000290140005)

Date:
Place: Ghaziabad

CERTIFICATE

Certified that **NITIKA PATEL (University Roll No: 2000290140078) ABHINAV SHARMA (University Roll No: 2000290140005)** have carried out the project work having “**COLLEGE ADMIN MANAGEMENT SYSTEM**” for Master of Computer Applications from **Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow** under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

NITIKA PATEL (2000290140078)

ABHINAV SHARMA (2000290140005)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Mr. S.D.Mishra

Associate Professor

**Department of Computer Applications
KIET Group of Institutions, Ghaziabad**

Signature of Internal Examiner

Signature of External Examiner

Dr. Ajay Shrivastava

**Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad**

ABSTRACT

This is a web oriented application allows us to access the whole information about the college, staffs, students, facilities etc. This application provides a virtual tour of Campus. Here we will get the latest information about the students and staffs. This generic application designed for assisting the students of an institute regarding information on the courses, exams, grades and exam-timetable. It also provides support that a faculty can also check about his daily schedule, can create exams, prepare questions for Exams, and notices to the students. Here administrator will manage the accounts of the student and faculties, checks the overall performances of students, and upload the latest information about the campus.

We identify several problems including unauthorized privilege escalation, incorrect use of cryptography, vulnerabilities to network threats, and poor software development processes. We show that only college administrator can start the system administrator can search the particular student by his/her enrollment number or student id and we are adding notification module where administrator should add the notification with start and end date.

ACKNOWLEDGEMENT

I express my deep sense of gratitude to Mr. Vidhushi. Mishra, Associate Professor **Department of Computer Applications KIET Group of Institutions, Ghaziabad**. Whose kindness valuable guidance and timely help encouraged me to complete this volume on a very crucial issue related to the work and who helped me in completing the project and he exchanged his interesting ideas, thoughts and made this project easy and accurate.

I express my thanks to **Mr. Prashant Agrawal and Ms. Neelam Rawat** for extending his support.

I express my thanks to **Dr. Ajay Shrivastav, Head, Department of Computer Applications KIET Group of Institutions, Ghaziabad**.for kindly intention.

I express my thanks to the authors whose works have been consulted by me during the project.

I would also thank my institution and my faculty members without whom this project would have been a distant reality.

I wish to thank my parents for their undivided support and interest who inspired me and encouraged me to go my own way without whom I would be unable to complete my project .

At last but not the least I want to thank my friends who appreciated me for my work and motivated me and finally to god who made all the things possible.

NITIKA PATEL (2000290140078)
ABHINAV SHARMA (20002901400)

CHAPTER - 1

INTRODUCTION

PROJECT DESCRIPTION

For accomplishing big projects, the number of developers works collectively on different modules. Their efforts when combined together gives the final outcome .However, a member working on one module may encounter the need to understand other modules. Hence, he may feel the need of telling the concerned member to explain his module. It may be time consuming and troublesome for the concerned member to explain the entire code of the module .Therefore, there arises a need for a tool like CLASS BROWSER which gives the class diagram of the entire module (project). It is quite reliable and easy to understand. It also helps in debugging large projects.

The traditional view of software development takes an algorithmic perspective. In this approach, the main building block of all software is the procedure or function. This view leads developers to focus on issues of control and the decomposition of larger algorithms into smaller ones. There is nothing inherently evil about such a point of view except that it tends to yield brittle systems. As requirements change and the system grows, systems built on algorithmic focus turn out to be very hard to maintain.

The contemporary view of software development takes an object-oriented perspective. In this approach, the main building block of all software systems is the object or class. Simply put, an object is a thing, generally drawn from the vocabulary of the problem space or the solution space, a class is a description of a set of common objects. Every object has identity, state, and behavior.

OBJECTIVE

This is a web oriented application allows us to access the whole information about the college, staffs, students, facilities etc. This application provides a virtual tour of Campus. Here we will get the latest information about the students and staffs. This generic application designed for assisting the students of an institute regarding information on the courses, exams, grades and exam-timetable. It also provides support that a faculty can also check about his daily schedule, can create exams, prepare questions for Exams, and notices to the students. Here administrator will manage the accounts of the student and faculties, checks the overall performances of students, and upload the latest information about the campus.

PROJECT SCOPE

- College information: Through this service one can access the complete information about the college campus such as courses available, admission procedure, placements, college events, achievements etc.
- Student tracking: Any company or any organization that want to check the summary about the student of the college, so that they will be able to choose the particular students for their campus placement And for that purpose they will be given a particular link through which they can access the information required.
- Student attendance status: It gives the attendance status of students. Faculty will update the attendance periodically and can be seen by students.
- Online Exams: Students can take online exams from anywhere; it provides portability and system for calculating their performance as well. Online Exam Dates will be provided by the Faculties to the Students Through their registered Email.

- Student's performance in exams: This facility provides the performance of the student in each exam which is conducted by university or college such as midterm performance. Marks obtained by students in exams will be automatically updated by system which can be accessed by students.
- Exam Notification: This facility notifies students and parents about examination schedule.
- Events: It will give information about different events that will be conducted by college time to time. Information about these events will be updated by administrator.
- Online assignments: This service provides the facility to faculty to upload assignments and to students to submit these assignments online.

Information about staff: It will help in maintaining complete information about college faculty members such as their department, profile etc. Administrator will register new faculties and remove their account when they leave the college.

Description of the project:

We identify several problems including unauthorized privilege escalation, incorrect use of cryptography, vulnerabilities to network threats, and poor software development processes. We show that only college administrator can start the system administrator can search the particular student by his/her enrollment number or student id and we are adding notification module where administrator should add the notification with start and end date.

INITIAL INVESTIGATION

The first phase of software project is to gather requirements .Gathering software requirements begins as a creative brainstorming process in which the goal is to develop an idea for a new and modules that no other software vendor has thought. New software modules ideas normally developed as a result of analyzing the project

The main function of requirements gathering phase is to take an abstract idea that fills a particular needs or that solves a particular problems and create a real world project with a particular sets of objectives, timeline and team.

Some of the highlights of the requirements gathering phase include:

- Collecting project ideas.
- Gathering customer requirements and proposed solution.
- Justifying the project.
- Submitting the request for proposal
- Getting the team in phase.
- Preparing the requirements documents.

Collecting project ideas:

coming up with project ideas can prove expansion exercise.

Problem Definition

The problem is to provide the complete information about the college campus. In which the college staff members and students can access the information and will be familiar with college campus. It will provide interactive environment for the staff and students by getting knowledge of attendance, remarks, exams performances, grades, exam-timetables, notices etc.

TECHNOLOGY SELECTION

- Advance Java
- Tomcat Server
- MySql Server
- Servlets
- Jsp
- HTML / CSS
- Bootstrap
- Javascript

We select the java language for our project, because Java is an object oriented programming language.

It has two lives, one as a stand-alone computer language for general-purpose programming and the other as a supporting language for Internet programming.

Advance Java is used for web development purpose by using it as a backend system programming language.

Java offers the following features:

1. Compiled and Interpreted.
2. Platform-Independent and Portable.
3. Object-Oriented.
4. Robust and Secure.
5. Distributed
6. Architecture-Neutral.
7. Familiar and Simple.
8. Multithreaded and Interactive.
9. High Performance.
10. Dynamic and Extensible

OVERVIEW OF ADVANCE JAVA

Java is divided into two parts i.e. Core Java (J2SE) and Advanced Java (JEE). The core Java part covers the fundamentals (data types, functions, operators, loops, thread, exception handling, etc.) of the Java programming language. It is used to develop general purpose applications. Whereas Advanced Java covers the standard concepts such as database connectivity, networking, Servlet, web-services, etc.

The four major benefits of advance Java that are, network centric, process simplification, and futuristic imaging standard.

- JEE (advance Java) provides libraries to understand the concept of **Client-Server architecture** for web- based applications.
- We can also work with web and application servers such as **Apache Tomcat** and **Glassfish** Using these servers, we can understand the working of HTTP protocol. It cannot be done in core Java.
- It is also important understand the advance Java if you are dealing with trading technologies like **Hadoop, cloud-native** and **data science**.
- It provides a set of services, **API** and **protocols**, that provides the functionality which is necessary for developing **multi-tiered** application, web-based application.
- There is a number of advance Java frameworks like, **Spring, Hibernate, Struts**, that enables us to develop secure **transaction-based** web applications such as banking application, inventory management application.
-

HARDWARE AND SOFTWARE USED IN THE PROJECT

PROCESSOR	INTEL CORE PROCESSOR OR BETTER PERFORMANCE
OPERATING SYSTEM	WINDOWS7, WINDOWS8,WINDOWS10, UBUNTU
MEMORY	1GB RAM OR MORE
HARD DISK SPACE	MINIMUM 3 GB FOR DATABASE USAGE FOR FUTURE
LANGUAGE	JAVA SERVER PAGE
IDE	NETBEANS 7.01
DATABASE	MS ACCESS

FRONT END - BACK END CONNECTIVITY

Html, Css, Javascript as Frontend

1. The letters in HTML stand for Hypertext Markup Language. HTML as a whole is the markup that creates the basic elements we view on a website.
2. Cascading Style Sheets, or CSS, is what gives our HTML visual appeal and draws in the user. To put it simply, style sheets dictate the presentation of HTML elements on a page.
3. JavaScript is a runtime language for web browsers. This means that when you open a web page, the page will load both the foundational and any new JavaScript

Java, MySQL as Back end

4. Java offers several benefits to the developer creating backend end application for database server.
5. Java is platform independent language hence can be hosted on architecture that support on JVM.
6. Java also reduces cost related to deployment & maintenance of hardware & software.
7. Java base servers are thinking like that uses minimum hardware resources.
8. There are big incentives to create java based solution for corporate as shifting their application across the architecture will not involve overhead or cost.

Databases are the systems that contain many different objects used together to facilitate fast and efficient access to the data.

MySQL is an application program interface form Microsoft that lets a programmer writing Windows applications; get access to a relational as well as non-relational database from both Microsoft and other database providers.



SYSTEM DESIGN

Purpose of **College ERP System for a College Design Document** is to describe the design and the architecture of **CES**. The design is expressed in sufficient detail so as to enable all the developers to understand the underlying architecture of **CES**. Logical architecture of **JDBC** driver, Server, **DML**, **DDL**, Session and Data Store are explained.

CES requires Java **JRE** 1.8 or higher. Since **CES** is written in Java, it can run on any platform that supports the Java runtime environment 1.8 or higher. The compiled files are contained in Java Archives (**JAR**'s) and have to be defined in the **CLASSPATH** environment variable.

Logical View:

It provides the user with an abstract view of the overall system functionality

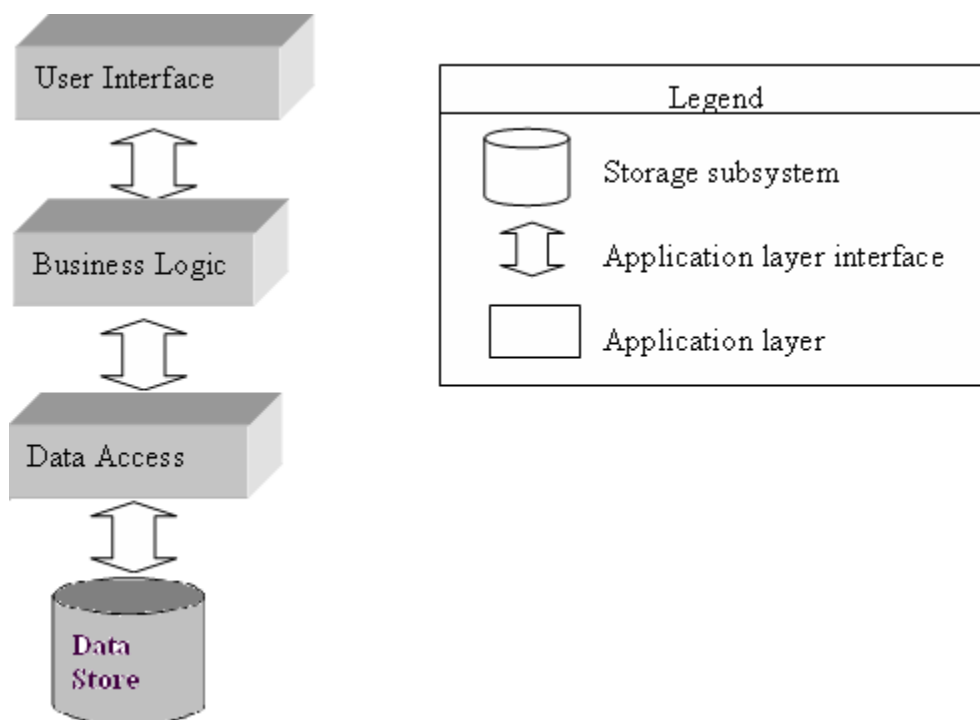


fig: Abstract view of CES (College ERP System)

EVALUATION OF GOOD DESIGN SOFTWARE

Scope and Boundaries

Scope of this system can be described as follows:-

- i. It is an online web app, available for desktop browsers used by admin, faculty, students to do different things.
- ii. The storing and retrieval of data is quick
- iii. It is an online application.
- iv. Access is given to only authorize person and no other person can access this Web app.
- v. Data security is maintained properly by authentication of users.
- vi. Large amount of data can be processed quickly with ease

User view

1. **Admin:** It is the administrator account having rights to govern whole web app. He can create and delete faculty accounts and details , create and change password, he also have rights to add and delete record, various criteria and notification.
2. **Faculty:** It is the Teacher account having rights to do specific tasks. He can create and delete students and details create and change password, he also have rights to add and delete record, various criteria and notification.
3. **Student:** Students have various permissions as well like taking online Exams, checking their results, attendance, modules, remaining fee etc.
4. **Fee Admin:** It is an admin account for submitting fee of students, checking students with remaining fee etc

CHAPTER - 2

FEASIBILITY STUDY

TECHNICAL,ECONOMICALLY& OPERATIONAL FEASIBILITY

Perform and evaluate feasibility studies like cost-benefit analysis, technical feasibility, time feasibility and operational feasibility for the project. Project Scheduling should be made using PERT charts.

Feasibility study is carried out to decide whether the proposed system is feasible for the company. The feasibility study is to serve as a decision document it must answer three key questions:

2. Is there a new and better way to do the job that will benefit the user?
3. What are the cost and the savings of the alternative(s)?
4. What is recommended?

Financial feasibility:

Financial feasibility refers to financial support required. It refers to finance incurred during the development of the project.

Technical feasibility:

Technical feasibility refers to technical knowhow and auxiliary devices required.

Behavioral feasibility:

Refers to reaction of the people towards the project.

Operational feasibility:

Operational feasibility means is it possible to practically implement the project. While installing this software, the hardware and software requirements should be specified.

FEASIBILITY GAINED BY OUR SYSTEM

Technical Feasibility

Since our project is in JAVA so we need to have a strong base in programming. A computer with JDK installed is needed.

Financial Feasibility

To implement the system we require an online Hosting Service. Since the system will be implemented on online platform so, there will be no need to buy the computers. The system is economically feasible to implement.

Operational Feasibility

Our system will be easy to install and use. Hence our system is operationally feasible. System is hosted online so anyone can access it with computer system and internet connectivity.

Cost-Benefit Analysis

The cost incurred by our system includes only the web hosting cost and cost of the computer needed to run the project. The benefits incurred by our system will include.

CHAPTER - 3

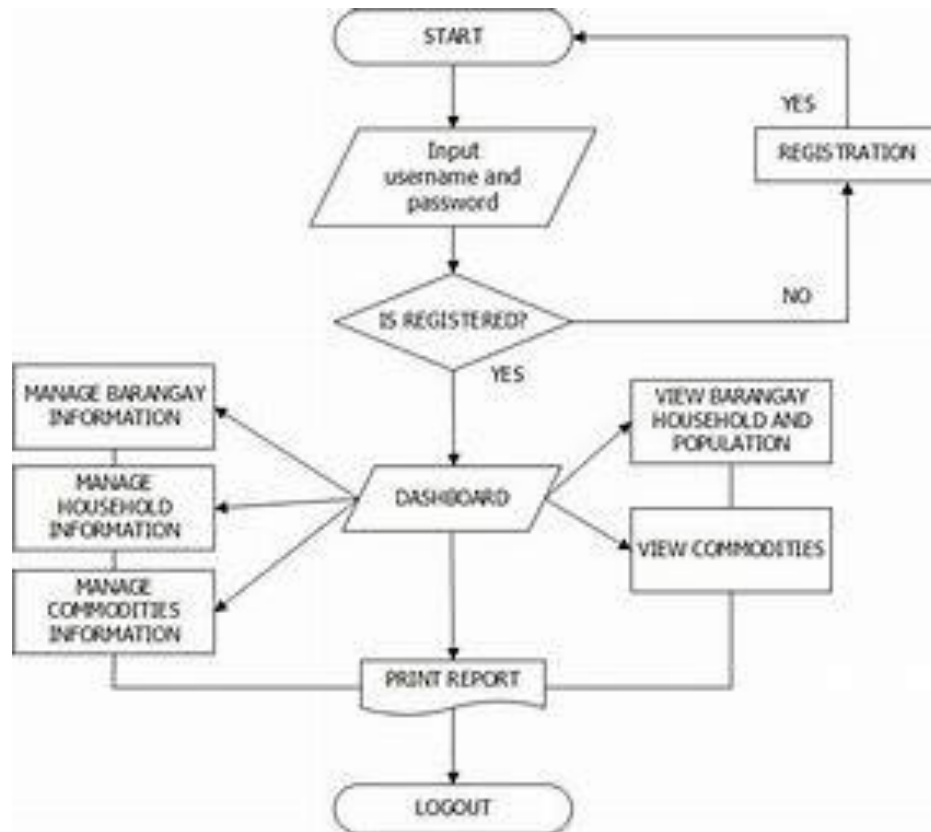
DATABASE DESIGN



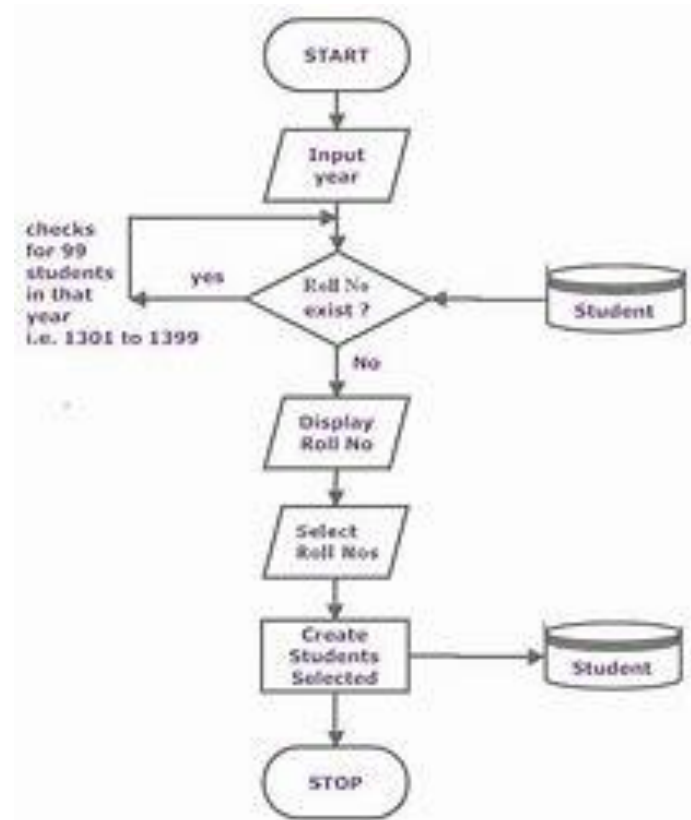
DATABASE TABLE

Table name	Purpose
1.admin	Admin table contain id, password and user name field.
2.teacher	Teacher table contain id, password and user name field of teachers.
3.fee admin	Fee admin table contain id, password and user name field of Fee Admin.
4.student	Student table contain student details like name, dob, address etc.
5.notification	Notification table contain notification detail.
6.result	Result table contain the performance details about the student.
7.exams	Exams table contains details of exams.
8.examQues	ExamQues table contain the questions of a particular exam.
9.department	Department table contains department name and id.

FLOW CHART



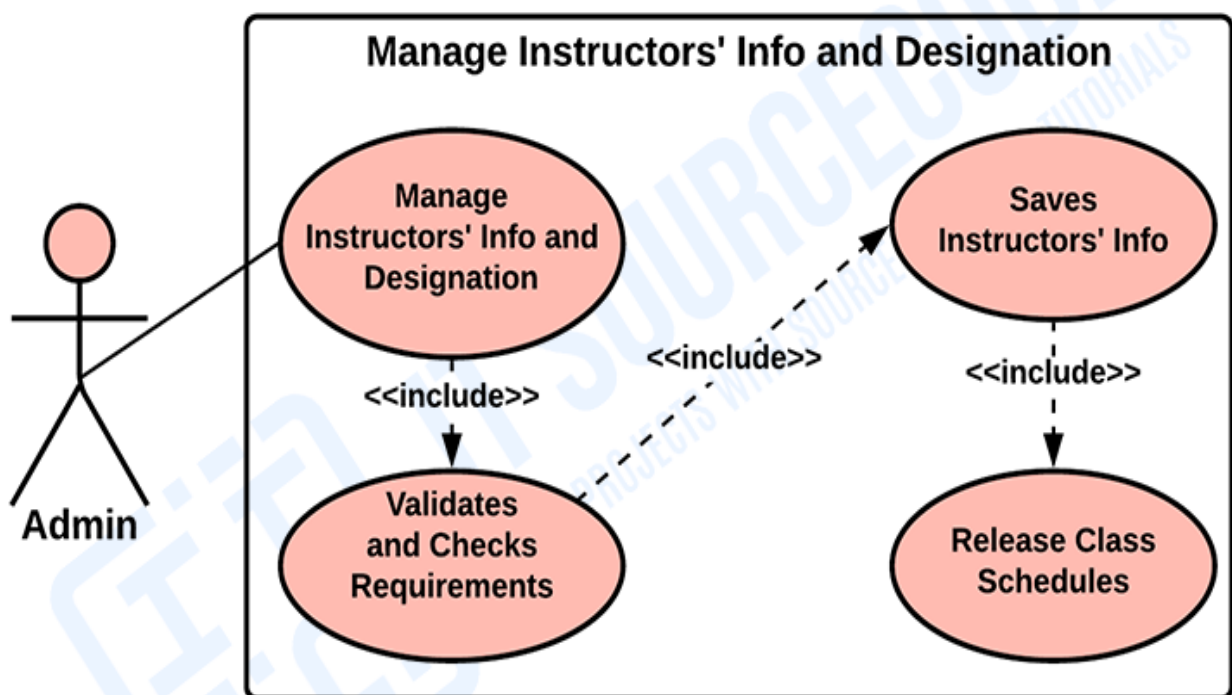
Fig(1) Administrator's system flow chart



Fig(2)flow chart for Creating Student by Admin

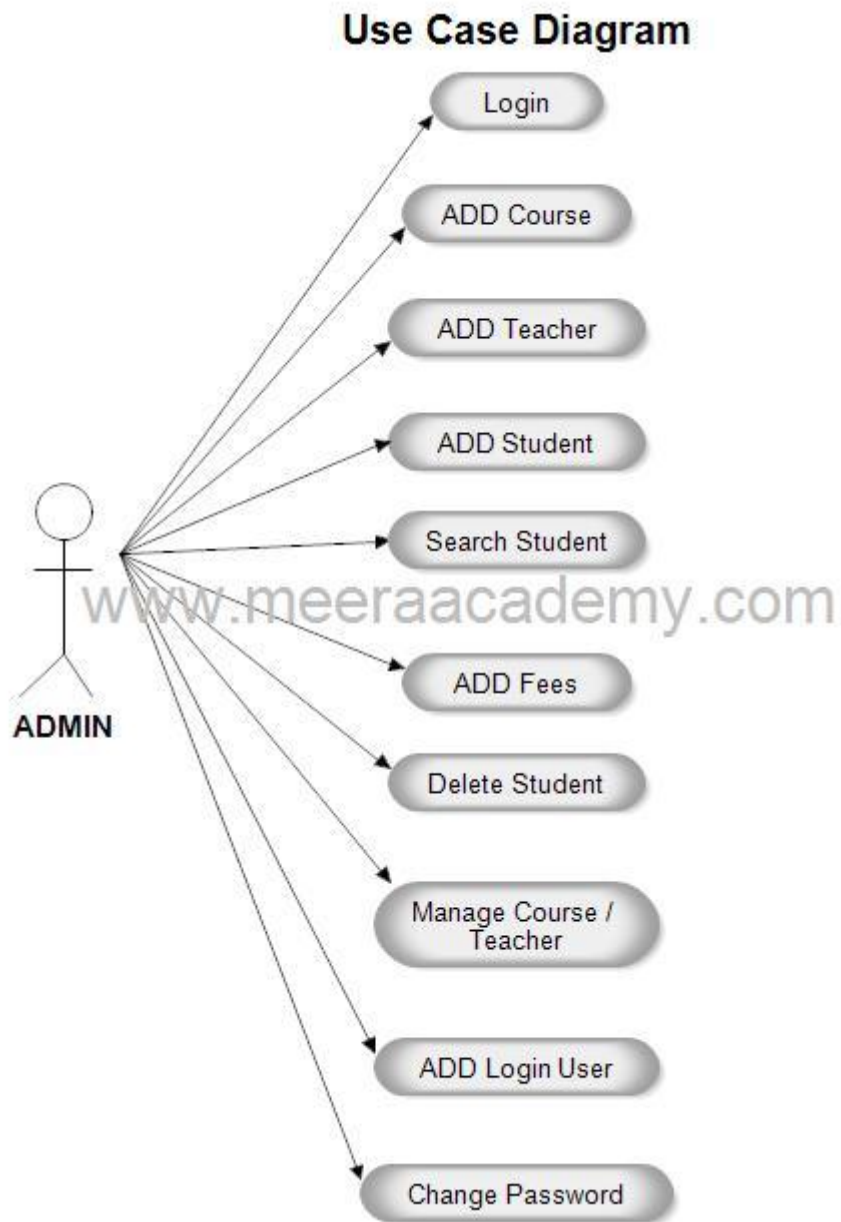
USE CASE DIAGRAM

COLLEGE MANAGEMENT SYSTEM

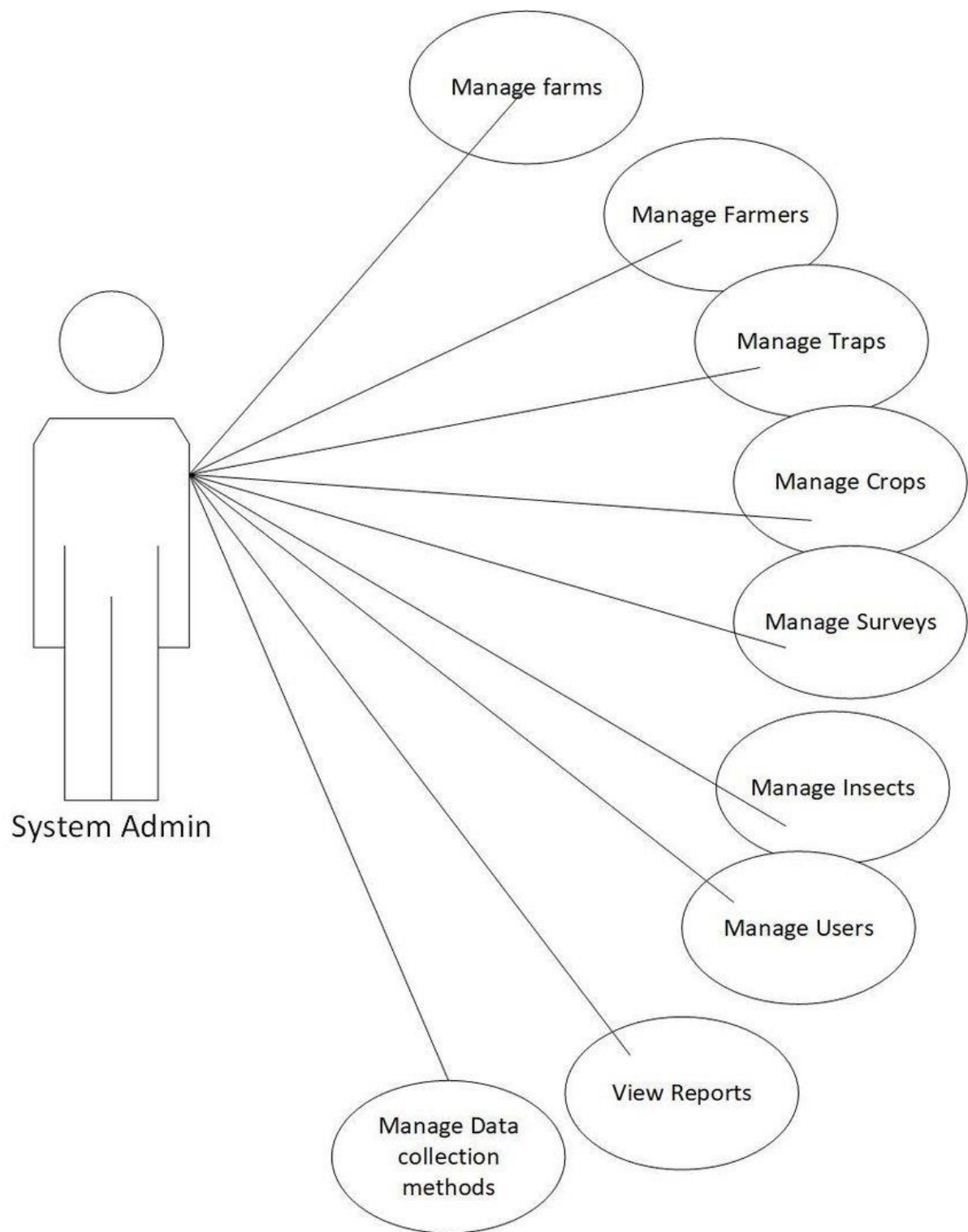


USE CASE DIAGRAM

Fig(3)

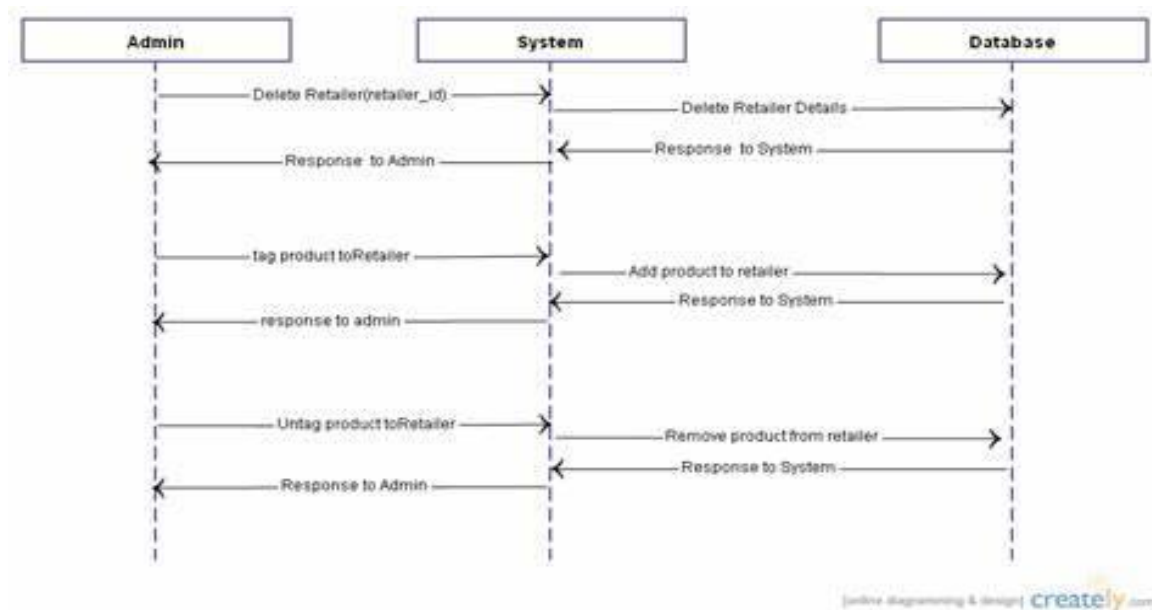


Fig(4) Student Information

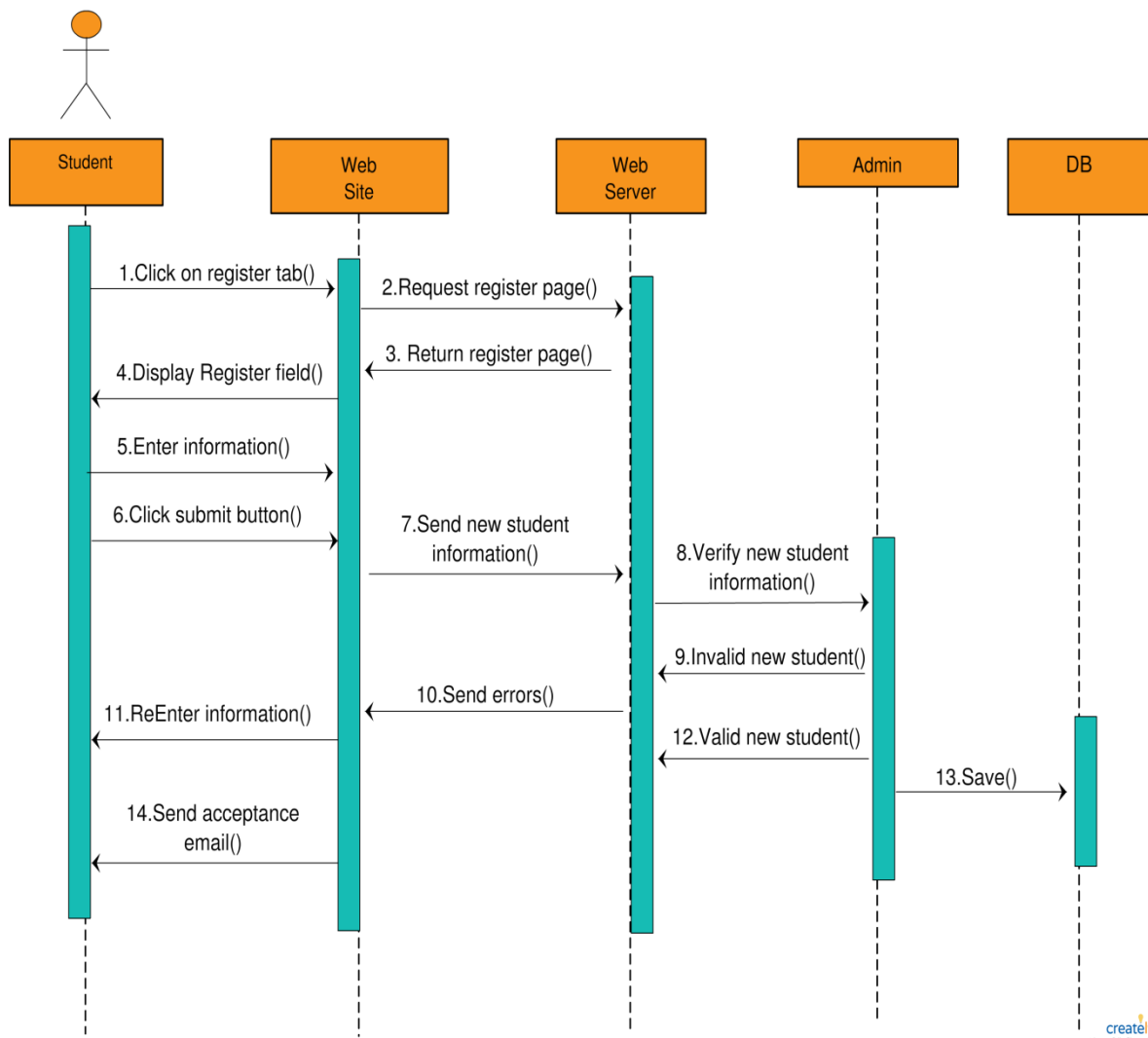


Fig(5)System Admin

SEQUENCE DIAGRAM



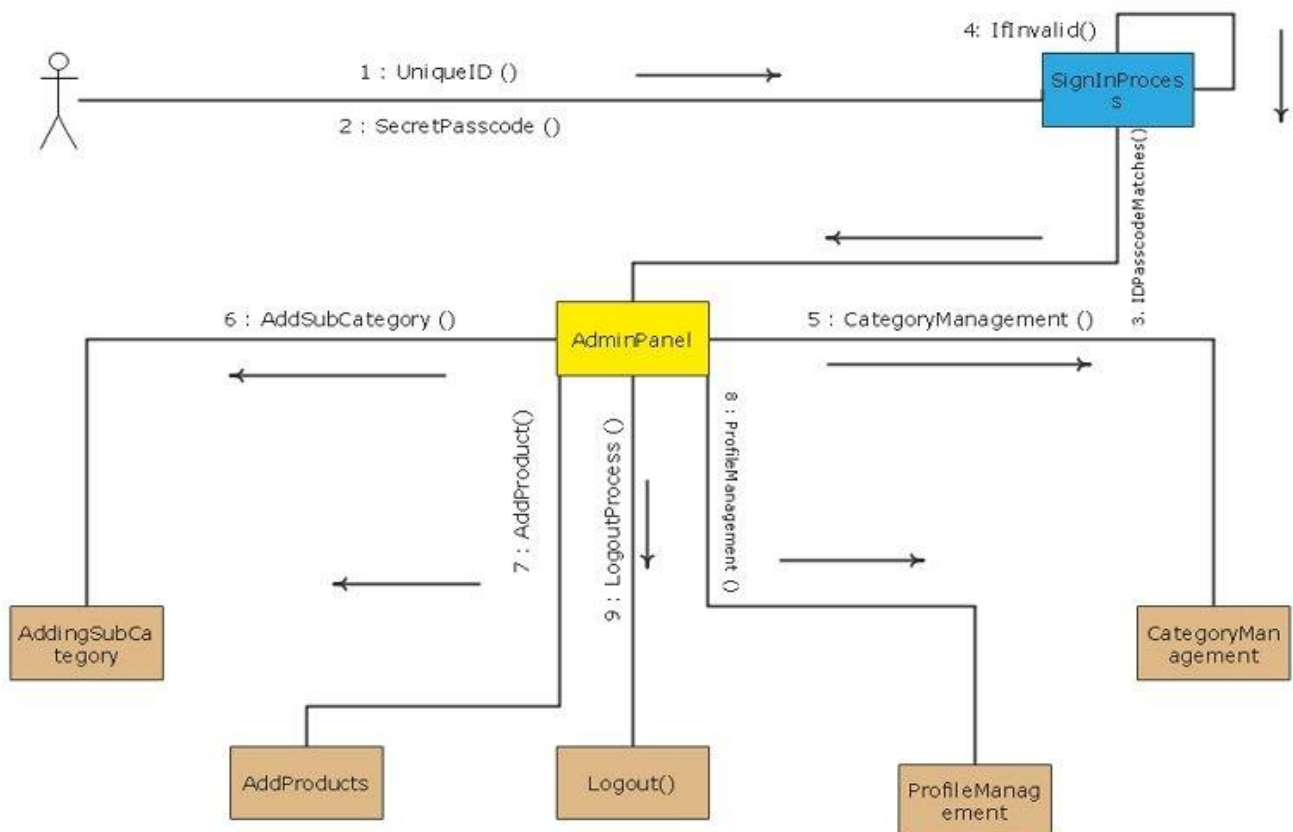
FIG(6)Admin



Fig(7) Student Resigtration

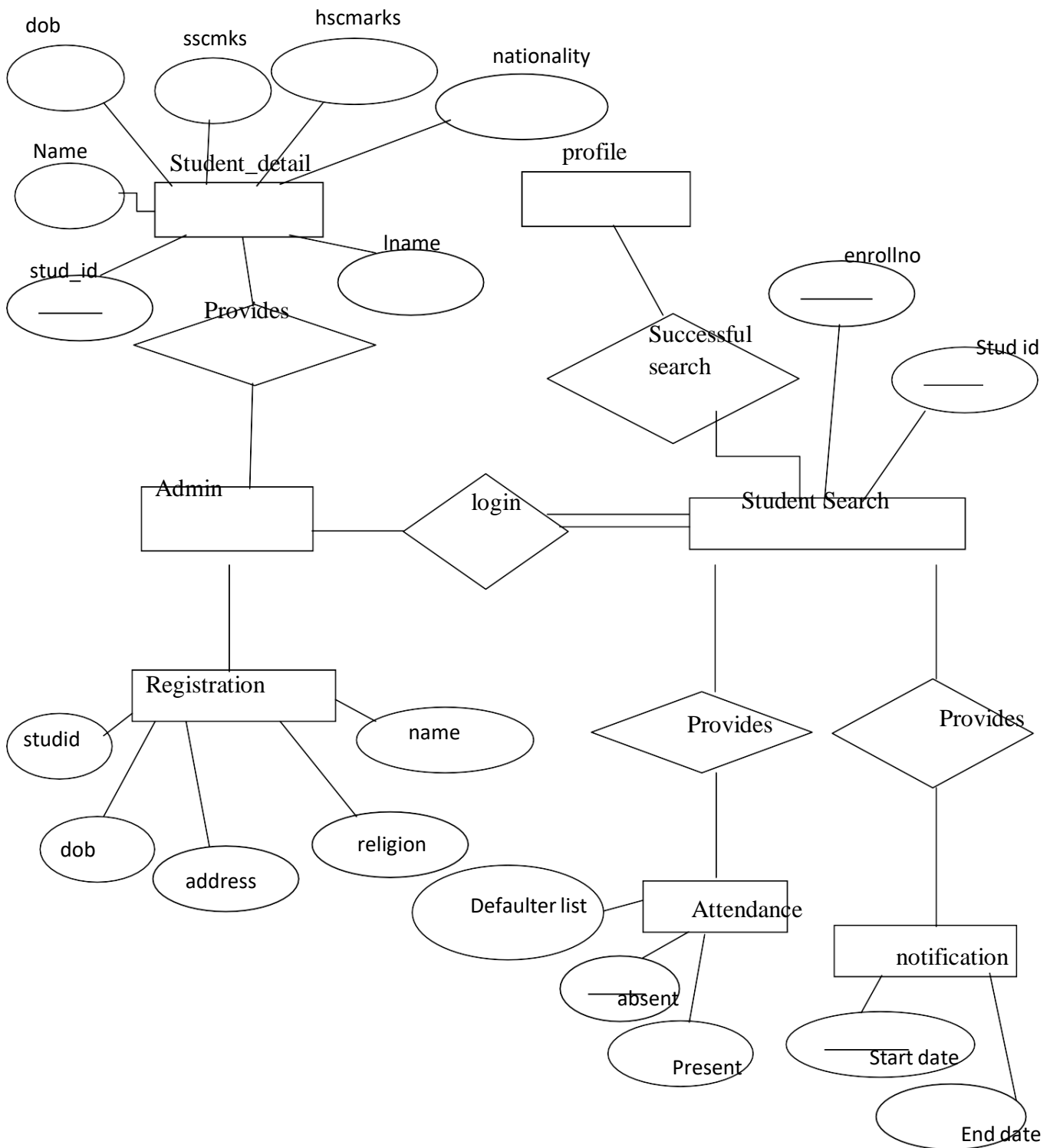
COLLABORATIVE DIAGRAM

**Collaboration Diagram
For AdminPanel**

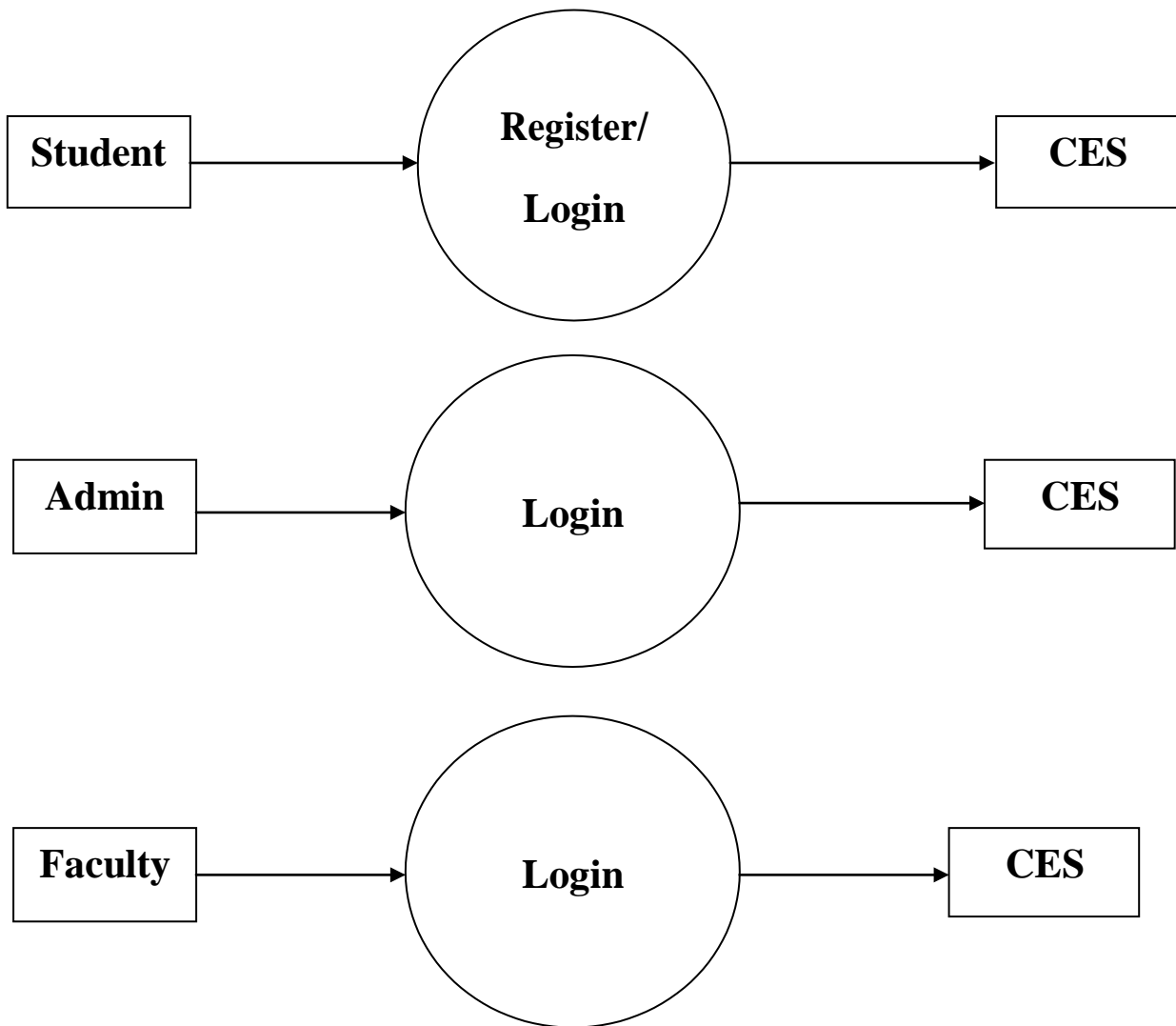


Fig(8)

ENTITY RELATIONSHIP DIAGRAM

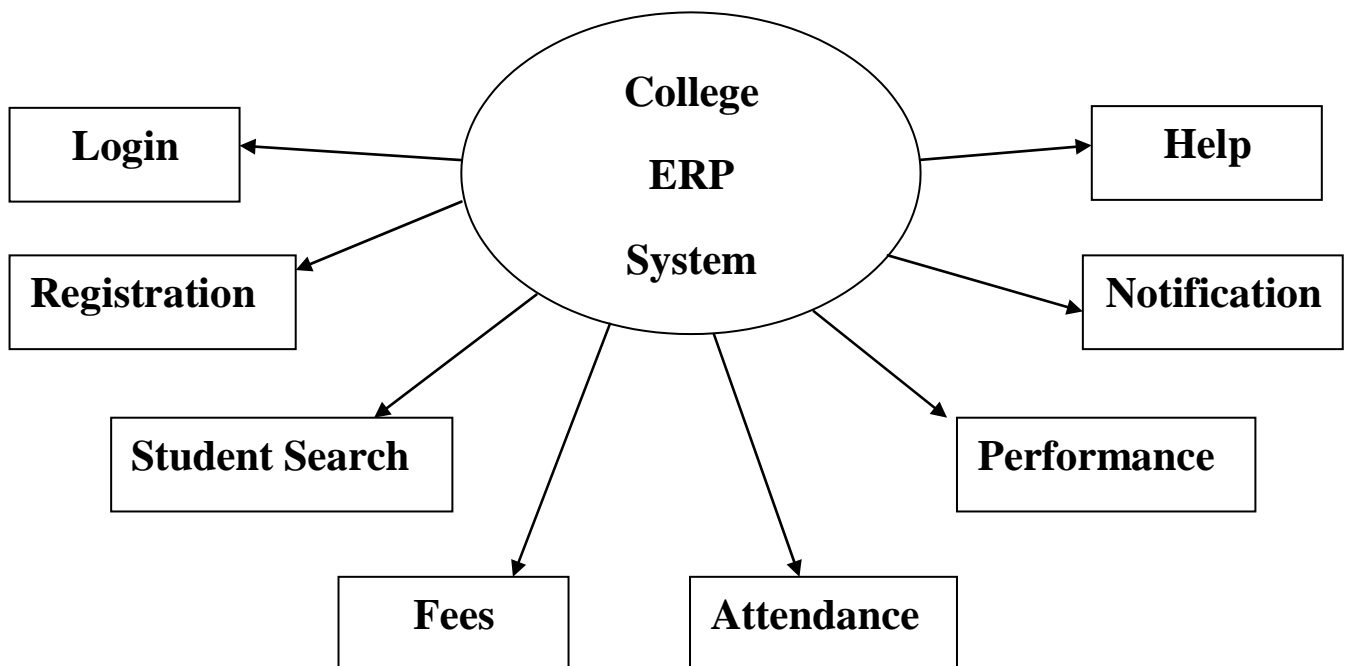


CONTEXT LEVEL DFD Level-0



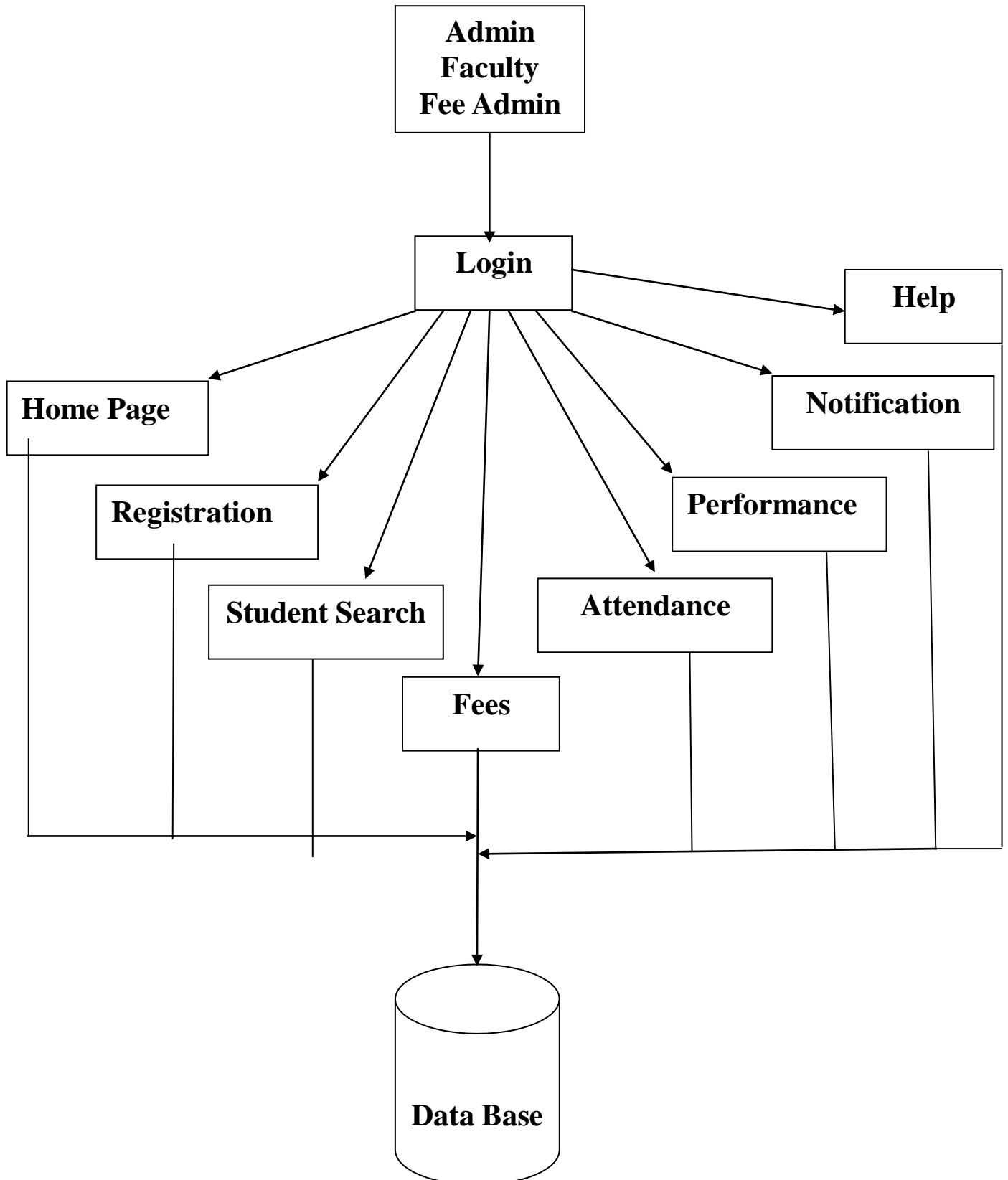
LEVEL- 1 DFD

Level-1



LEVEL- 2 DFD

Level -2



USER REQUIREMENTS:

The following requirements are raised during the analysis of the needs of the users:

- A Person should be able to login to the system through a login page of the Application.
- The Administrator can create users as per user requirement.
- Student can register themselves through a registration page. On successful registration of user, (Student/Faculty/Admin) can view reports.
- A general user will have access to see the status of particular Student id number.
- Student (user) can use all the facilities, same as which are provided to him in the college.
- Student can see attendance, notices, grades, report and other facilities in updated manner.
- There will be a separate page for every student as his account in which he can get notices, attendance, grades etc.
- Faculty can give the attendances and notices for the students.
- The administrator verifies all these reports and generates them for users to view them.
- Faculty can create online exam schedule and send the details to the students.

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

ANALYSIS MODEL

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.

- A second prototype is evolved by a fourfold procedure:
 1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
 2. Defining the requirements of the second prototype.
 3. Planning and designing the second prototype.
 4. Constructing and testing the second prototype.

- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.

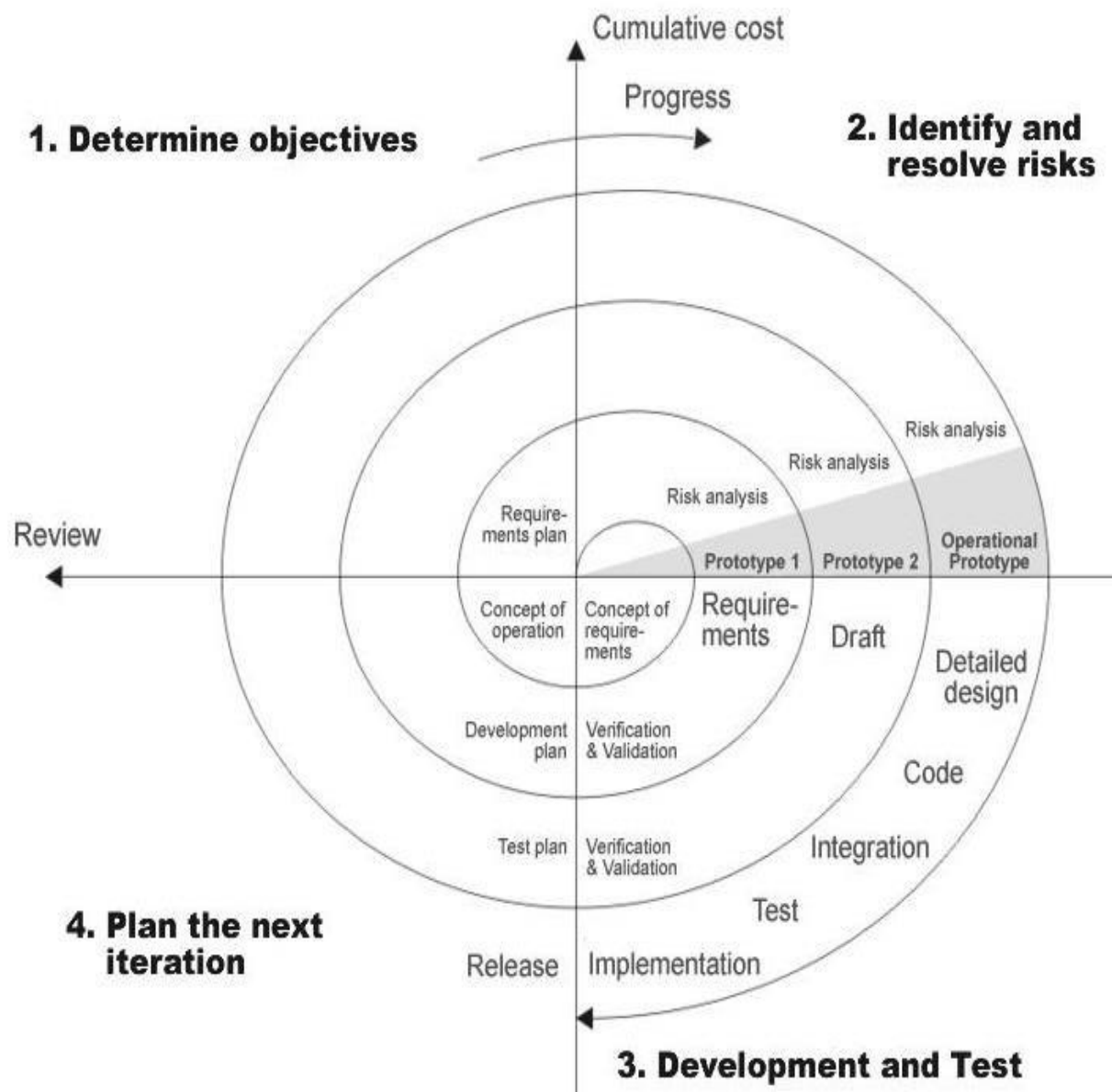
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.

- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.

- The final system is constructed, based on the refined prototype.

- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

The following diagram shows how a spiral model acts like:



STUDY OF THE SYSTEM

Graphical user interface

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI'S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Data updation along with the extensive data search capabilities.

The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

NUMBER OF MODULES

The system after careful analysis has been identified to be presented with the following modules

:

The modules involved are:

1. College information: Through this service one can access the complete information about the college campus such as courses available, admission procedure, placements, college events, achievements etc.

2. Student tracking: Any company or any organization that want to check the summary about the student of the college, so that they will be able to choose the particular students for their campus placement And for that purpose they will be given a particular link through which they can access the information required.
3. Student attendance status: It gives the attendance status of students. Faculty will update the attendance periodically and can be seen by students.
4. Student's performance in exams: This facility provides the performance of the student in each exam which is conducted by university or college such as midterm performance. Marks obtained by students in exams will be automatically updated by system that can be accessed by students.
5. Exam Notification: This facility notifies students about examination schedule.
6. Events: it will give information about different events that will be conducted by college time to time. Information about these events will be updated by administrator.
7. Online Modules: This service provides the facility to faculty to upload modules and to students to learn about subjects.
8. Information about staff: It will help in maintaining complete information about college faculty members such as their department, profile, etc. Administrator will register new faculties and remove their account when they leave the college.

CHAPTER - 4

**FORM DESIGN AND
ARCHITECTURE**

FORM DESIGNING

- **Login Form:-**

This is login form of the web app, it asks for the id and password of the user, access is given to the authorized persons only. Types of logins are there Admin, Faculty, Fee Admin and Student.

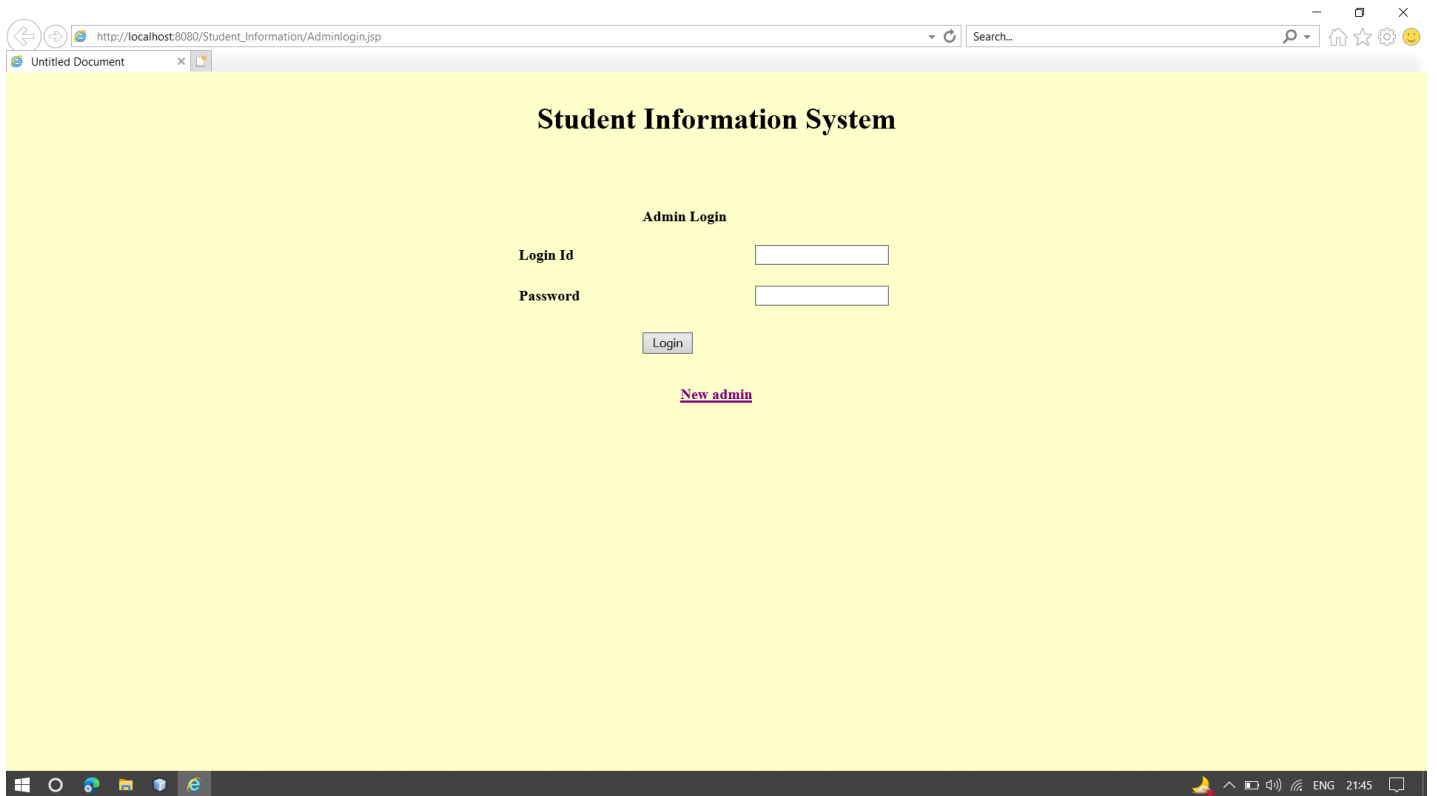
- **Index Page:**

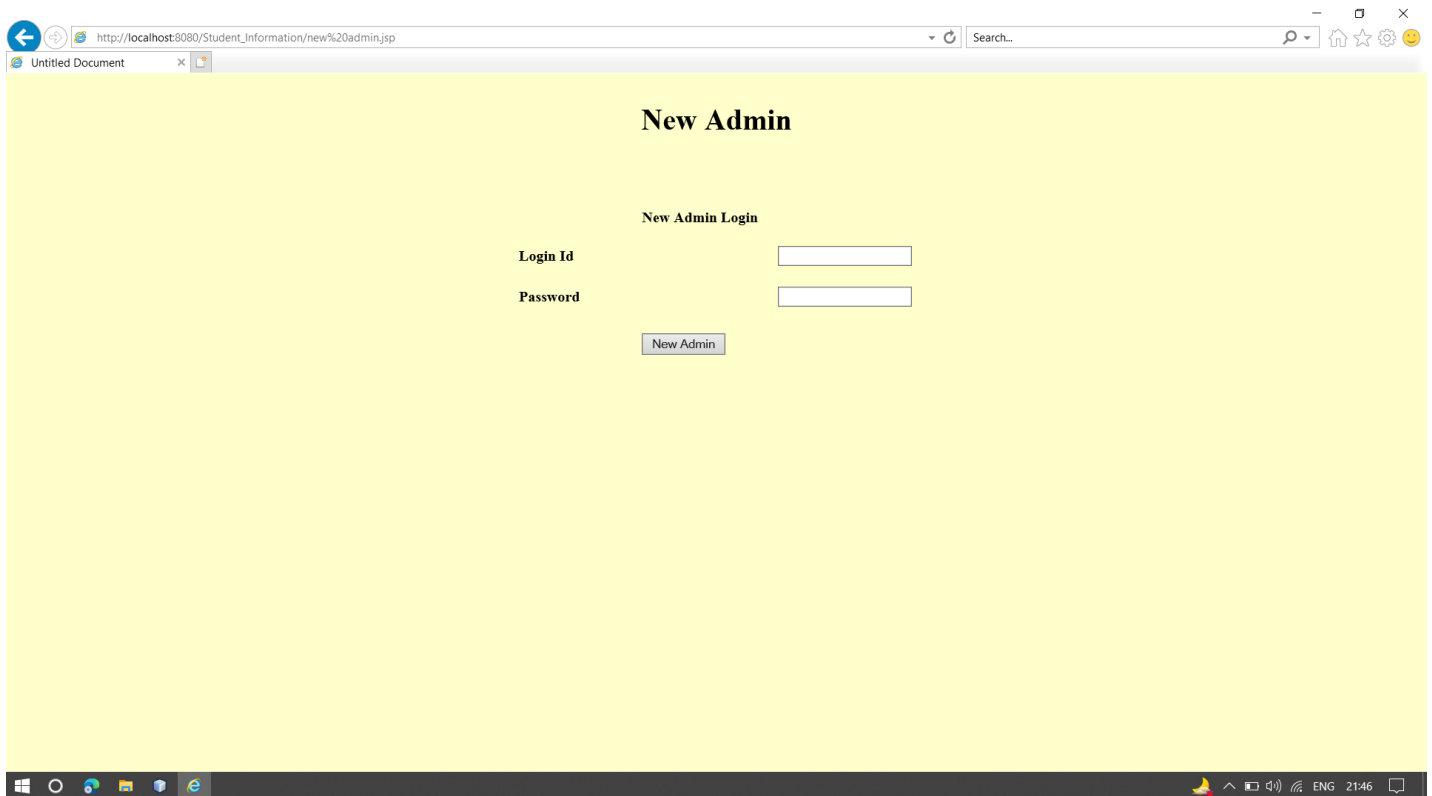
This screen acts as the home screen for this web app. Through this we can access its login forms and registration forms. This page gives the user further 5 options:-

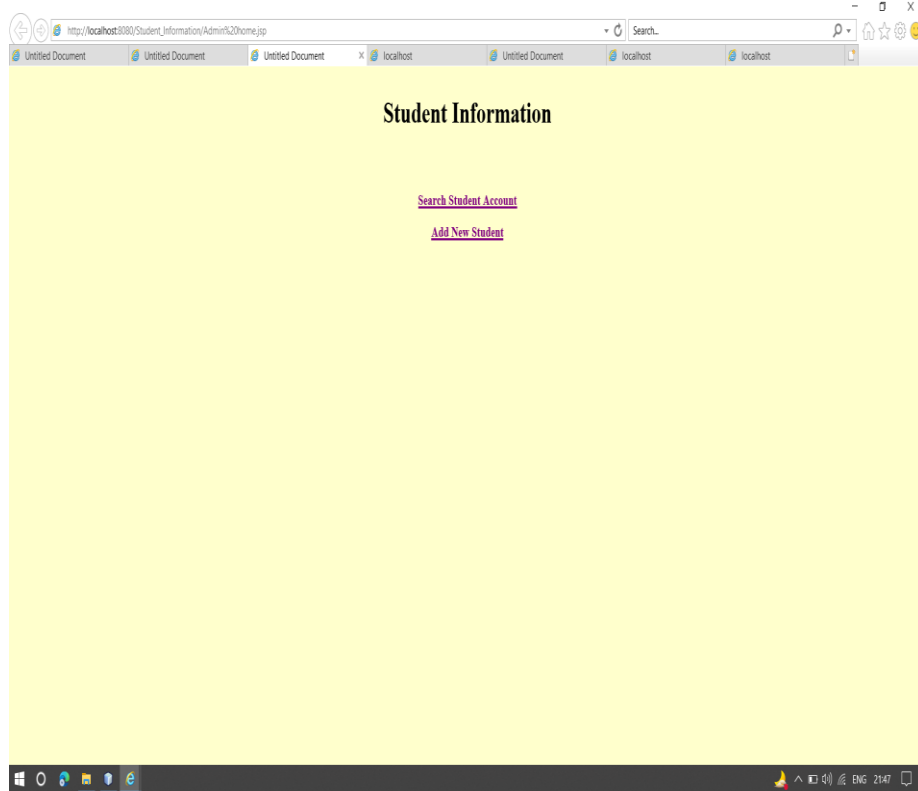
- **Login Page:** It contains LOGIN, LOG OUT and DASHBOARD option to do specific tasks of particular user.
- **Registration Page:** This module is designed for registrations

- **Form list and details**

1. **Registration Entry Form:** registration entry form can be used to enter the new entry of student, faculty, exams, questions etc.
2. **Student Search Form:** student search form can be used by admin, faculty, for searching particular student data.
3. **Notification Form:** notification form can be used to add notification related to the exams and schedule of the college. Full authority is given to the admin.
4. **Fees Form:** this can be generated the fees report of the students.





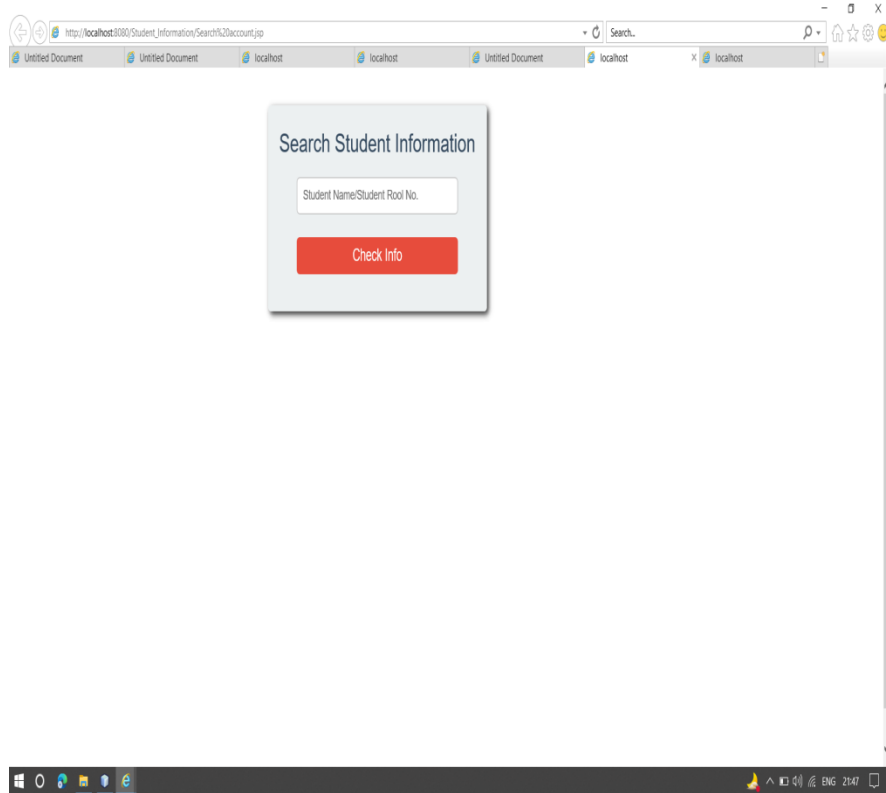


Browser window showing a search interface for student information. The URL is `http://localhost:8080/Student_Information/Search%20account.jsp?un=nitika`.

The search interface includes a text input field labeled "Student Name/Student Rool No." and a red button labeled "Check Info".

Below the search interface, a table displays the retrieved student information:

Name	: nitika
Father name	: narendra
Email id	: nitikapatel34@gmail.com
Branch name	: mca
Branch name	: 4
Semester	: 25
Rool no	: 02/02/2021
Open_date	: xyz
Address	: 12367



Browser window showing a web application running on localhost:8080/Student_Information/Newaccount.jsp.

Student Information

New Account

Name	<input type="text"/>
Father name	<input type="text"/>
Email id	<input type="text"/>
Branch name	<input type="text"/>
Semester	<input type="text"/>
Rool no	<input type="text"/>
Open_date	<input type="text"/>
Address	<input type="text"/>
Phone_no	<input type="text"/>

Submit

CHAPTER – 5

CODING

Admin home page

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
.style1 {
    font-size: xx-large;
    font-weight: bold;
}
body {
    background-color: #FFFFCC;
}
-->
</style>
</head>

<body>
<p align="center" class="style1">Student Information </p>
<center>
<p>&nbsp;</p>
<p><strong><a href="Search account.jsp">Search Student Account </a></strong></p>
<p><strong><a href="Newaccount.jsp">Add New Student</a></strong></p>
</center>
<p align="center" class="style1">&nbsp;</p>
<p class="style1">&nbsp;</p>
<p class="style1">&nbsp;</p>
```

Admin Login

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
.style1 {
    font-size: xx-large;
    font-weight: bold;
}
body {
    background-color: #FFFFCC;
}
.style2 {font-size: medium; font-weight: bold; }
-->
</style>
</head>

<body>
<p align="center" class="style1">Student Information System</p>
<form action="Userloginconnection.jsp">
<center>
<p>&nbsp;</p>
<table width="433" height="187" border="0">
<tr>
<td width="127">&nbsp;</td>
<td width="115"><strong>Admin Login </strong></td>
<td width="169">&nbsp;</td>
</tr>
<tr>
<td><strong>Login Id </strong></td>
<td>&nbsp;</td>
<td><input type="text" name="t1" required/></td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>&nbsp;</td>
<td><input type="text" name="t2" required/></td>
</tr>
<tr>
<td height="54">&nbsp;</td>
<td><input type="submit" name="Submit" value="Login" /></td>
<td><a href="new admin.jsp"></a></td>
</tr>
</table>
```

```
</center>
</form>
<p align="center" class="style2"><a href="new admin.jsp">New admin</a> </p>
<p class="style1">&nbsp;</p>
<p class="style1">&nbsp;</p>
</body>

</html>
```

NewAccount Login

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
.style1 {
    font-size: xx-large;
    font-weight: bold;
}
body {
    background-color: #FFFFCC;
}
-->
</style>
</head>

<body>
<p align="center" class="style1">Student Information </p>
<center><form method="post" action="Newaccountconnection.jsp" >
<p class="style1">New Account</p>
<table width="394" border="0">
<tr>
<td width="163"><strong>Name </strong></td>
<td width="7">&nbsp;</td>
<td width="202"><input type="text" name="t1" required/></td>
</tr>
<tr>
<td><strong>Father name </strong></td>
<td>&nbsp;</td>
<td><input type="text" name="t2" required/></td>
</tr>
<tr>
<td><strong>Email id </strong></td>
<td>&nbsp;</td>
<td><input type="text" name="t3" required/></td>
</tr>
<tr>
<td><strong>Branch name </strong></td>
<td>&nbsp;</td>
<td><input type="text" name="t4" required/></td>
</tr>
<tr>
<td><strong>Semester</strong></td>
<td>&nbsp;</td>
<td><input type="text" name="t5" required/></td>
```

```

</tr>
<tr>
  <td><strong>Rool no </strong></td>
  <td>&nbsp;</td>
  <td><input type="text" name="t6" required/></td>
</tr>
<tr>
  <td><strong>Open_date</strong></td>
  <td>&nbsp;</td>
  <td><input type="text" name="t7" required/></td>
</tr>
<tr>
  <td><strong>Address</strong></td>
  <td>&nbsp;</td>
  <td><input type="text" name="t8" required/></td>
</tr>
<tr>
  <td height="24"><strong>Phone_no</strong></td>
  <td>&nbsp;</td>
  <td><input type="text" name="t9" required/></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td><input type="submit" name="Submit" value="Submit" /></td>
  <td>&nbsp;</td>
</tr>
</table>
<p class="style1">&nbsp;</p>
</form>
</center>
</body>

</html>

```

NewAccountConnection

<%--

Document : reg

Created on : May 15, 2014, 12:41:43 PM

Author : prafull

--%>

<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>

<!DOCTYPE html>

<%

try

{

Class.forName('sun.jdbc.odbc.JdbcOdbcDriver');

Connection con=DriverManager.getConnection("jdbc:odbc:Studentsource");

String N=request.getParameter("t1");

String P=request.getParameter("t2");

String A=request.getParameter("t3");

String B=request.getParameter("t4");

String C=request.getParameter("t5");

String D=request.getParameter("t6");

String E=request.getParameter("t7");

String F=request.getParameter("t8");

String G=request.getParameter("t9");

String sql="insert into

Studentinfo(Name,Father_name,Email_id,Branch_name,Semester,Rool_no,Open_date,Address,Phone_no)values('"+N+"','"+P+"','"+A+"','"+B+"','"+C+"','"+D+"','"+E+"','"+F+"','"+G+"')";

int action=con.createStatement().executeUpdate(sql);

con.setAutoCommit(true);

if(action>=1)

{

out.println("Deatils is Saved Succesfully and you can search your details in search customer Option");

out.println("
");

out.println("Search account");

}

else

out.println("record not save.....");

```
}  
catch(Exception ex)  
{  
  
    out.println(ex.getMessage());  
}  
%>
```


SearchAccount

```
<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>

<html>
<head>
<title></title>

    <meta name="description" content="">
    <meta name="viewport" content="width=device-width">
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<link href="style.css" rel="stylesheet" type="text/css">
<link rel="stylesheet" type="text/css" href="engine1/style.css" />
<script type="text/javascript" src="engine1/jquery.js"></script>
</head>
<body>

    <jsp:include page="header.jsp"/>

    <div class="login-wrap">
        <h2>Search Student Information</h2>
        <div class="form">
            <form action="">
                <input type="text" placeholder="Student Name/Student Rool No. " name="un" width="200" />
                <button> Check Info </button>
            </form>
        </div>
    </div>
</div>

<%
try
{
    Class.forName('sun.jdbc.odbc.JdbcOdbcDriver');
    Connection con=DriverManager.getConnection("jdbc:odbc:Studentsource");

    Statement st=con.createStatement();
    String s="select * from Studentinfo where Name='"+request.getParameter("un")+"'OR
Rool_no='"+request.getParameter("un")+"'";
    ResultSet rs=st.executeQuery(s);

    %>
```

```

<%
while(rs.next())
{

%>

<div class="table">
<table border="2" align="center">

<tr>
<th align="right"><strong>Name</strong></th>
<td width="20" align="center">:</td>
<td width="200"><%=rs.getString("Name") %></td>
</tr>

<tr>
<th align="right"><strong>Father name </strong></th>
<td width="20" align="center">:</td>
<td width="100"><%=rs.getString("Father_name") %></td>
</tr>

<tr>
<th align="right"><strong>Email id </strong></th>
<td width="20" align="center">:</td>
<td width="100"><%=rs.getString("Email_id") %></td>
</tr>

<tr>
<th align="right"><strong>Branch name </strong></th>
<td width="20" align="center">:</td>
<td width="100"><%=rs.getString("Branch_name") %></td>
</tr>

<tr>
<th align="right">Branch name</th>
<td width="20" align="center">:</td>
<td width="100"><%=rs.getString("Semester") %></td>
</tr>

<tr>
<th align="right"><strong>Semester</strong></th>
<td width="20" align="center">:</td>
<td width="100"><%=rs.getString("Rool_no") %></td>
</tr>

<tr>
<th align="right"><strong>Rool no </strong></th>
<td width="20" align="center">:</td>
<td width="100"><%=rs.getString("Open_date") %></td>

```

```

</tr>

<tr>
  <th align="right"><strong>Open_date</strong></th>
  <td width="20" align="center">:</td>
  <td width="100"><%=rs.getString("Address") %></td>
</tr>

<tr>
  <th align="right"><strong>Address</strong></th>
  <td width="20" align="center">:</td>
  <td width="100"><%=rs.getString("Phone_no") %></td>
</tr>
</table>
<div>
  <%
}
%>

</table>
<%
}
catch(Exception ex)
{
  out.println(ex.getMessage());
}
%>
</div>
</body>
</html>

```

SignUpConnection

```
<%--
  Document   : reg
  Created on : May 15, 2014, 12:41:43 PM
  Author    : prafull
--%>

<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>
<!DOCTYPE html>
<%
try

    {

        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con=DriverManager.getConnection("jdbc:odbc:Studentsource");

        String N=request.getParameter("t1");
        String P=request.getParameter("t2");

        String sql="insert into Admin(Adminid>Password)values('"+N+"','"+P+"')";

        int action=con.createStatement().executeUpdate(sql);
        con.setAutoCommit(true);

        if(action>=1)
        {
            out.println("Deatils is Saved Succesfully");
            out.println("<br>");
            out.println("<a href='Adminlogin.jsp'>GO TO HOME PAGE FOR LOGIN</a>");
        }
        else
            out.println("record not save.....");
        }
    catch(Exception ex)
        out.println(ex.getMessage());
    }
%>
```

UserloginConnection

```
<%--
    Document   : aa.jsp
    Created on : Apr 7, 2014, 3:18:51 PM
    Author    : prafull
--%>

<%@page contentType="text/html" pageEncoding="UTF-8" import="java.sql.*"%>
<!DOCTYPE html>
<%
try
    {

        Class.forName('sun.jdbc.odbc.JdbcOdbcDriver');
        Connection con=DriverManager.getConnection("jdbc:odbc:Studentsource");

        Statement st=con.createStatement();
        String s="select * from Admin where Adminid='"+request.getParameter("t1")+"'and
Password='"+request.getParameter("t2")+"'";

        ResultSet rs=st.executeQuery(s);

        if(rs.next())
        {

            out.println("<a href='Admin home.jsp'>continue</a>");
        }
        else
        {
            out.println("PLEASE ENTER VALID ID AND PASSWORD.....");

            out.println("<a href='Adminlogin.jsp'>Try Again</a>");
        }

    }
catch(Exception ex)
    {

        out.println(ex.getMessage());
    }
%>
```

Header

```
<html>
<head>
<title></title>
  <meta name="description" content="">
  <meta name="viewport" content="width=device-width">
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

  <link href="style.css" rel="stylesheet" type="text/css">

</head>
<body>
</body>
</html>
```

NewAdmin

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
.style1 {
    font-size: xx-large;
    font-weight: bold;
}
body {
    background-color: #FFFFCC;
}
-->
</style>
</head>

<body>
<p align="center" class="style1">New Admin </p>
<center>
    <form method="post" action="Signupconnection.jsp" >
    <p>&nbsp;</p>
    <table width="433" height="187" border="0">
    <tr>
    <td width="129">&nbsp;</td>
    <td width="143"><strong>New Admin Login </strong></td>
    <td width="147">&nbsp;</td>
    </tr>
    <tr>
    <td><strong>Login Id </strong></td>
    <td>&nbsp;</td>
    <td><input type="text" name="t1" required/></td>
    </tr>
    <tr>
    <td><strong>Password</strong></td>
    <td>&nbsp;</td>
    <td><input type="text" name="t2" required/></td>
    </tr>
    <tr>
    <td height="54">&nbsp;</td>
    <td><input type="submit" name="Submit2" value="New Admin" /></td>
    <td>&nbsp;</td>
    </tr>
    </table></form>
</center>
<p align="center" class="style1">&nbsp;</p>
```

<p class="style1"> </p>
<p class="style1"> </p>

Style.css

```
*{padding:0; margin:0;}
body{font-family: 'Pontano Sans', sans-serif; font-size:14px; color:black; background:#ffffff;}
img{border:none;}
h1{ font-size:25px; color:#243036; font-family: 'Lobster Two', cursive; font-weight:normal;}
h2{ font-size:16px; color:#66a7c5; font-family: 'Pontano Sans', sans-serif; font-weight:normal;}
P{ line-height:20px; }
a{text-decoration:none; color:#66a7c5;}
a:hover{color:#ee3233;}
```

.header-warp{

```
background:#ee3233; padding-top:85px; padding-bottom:40px; overflow:hidden;
```

```
}
```

.logo h1{

```
float:left; color:#ffffff; font-size:45px; margin-left:30px; margin-top:-30px;
```

```
}
```

.logo-img{

```
float:left; margin-left:1px; margin-top:-55px;
```

```
}
```

.menu{ float:right;}

```
.menu ul li{ list-style:none; float:left; font-size:18px; padding:10px;}
```

```
.menu ul li a{color:#ffffff; text-decoration:none; padding:10px;}
```

```
.menu ul li a:hover{ background:#66a7c5; padding:10px;}
```

```
.menu ul li a.active{ background:#66a7c5; padding:10px;}
```

.slide{

```
margin-top: 100px;
```

```
margin-right: -300px;
```

```
float: right;
```

```
}
```

.slide div{

```
margin-bottom:20px;
```

```
}
```

#footer{

```
height: 170px;
```

```

        text-align: center;
        margin-top: 200px;
        background-color: #243036;
        color: #9b9e9a;
        font-size: 12px;
    }
    #footer img{
        border-radius: 10px;
        margin-top: 10px;
    }

```

```

.button1{
    margin-top: 0px;
    margin-left: 0px;

    padding: 40px; overflow: hidden;
}

```

```

p {
    margin: 50px 0;
}

```

```

.button {
    background-color: #bc1815;
    border: medium none;
    border-radius: 3px;
    box-shadow: 0 -3px 1px rgba(0, 0, 0, 0.45) inset, 0 2px 2px rgba(0, 0, 0, 0.25);
    color: #fff;
    margin-right: 25px;
    padding: 8px 14px 10px;
}

```

```

    text-shadow: 1px 1px 0 rgba(0, 0, 0, 0.5);
}

```

```

a {
    text-decoration: none;
}

```

```

.button:active {
    box-shadow: 0 -3px 1px rgba(255, 255, 255, 1) inset, 0 0 3px rgba(0, 0, 0, 0.9) inset;
    position: relative;
    top: 3px;
}

```

```

.button:active:after {
    background: none repeat scroll 0 0 #fff;
    bottom: -1px;
    content: "";
    height: 3px;
    left: 0;
}

```

```

    position: absolute;
    width: 100%;
}
.button.blue {
    background: none repeat scroll 0 0 #22c3eb;
}
.button.green {
    background: none repeat scroll 0 0 #67b600;
}
.button.orange {
    background: none repeat scroll 0 0 #da8a00;
}
.button.yellow {
    background: none repeat scroll 0 0 #b3a400;
}
.button.turquoise {
    background: none repeat scroll 0 0 #00b1af;
}
.button:last-of-type {
    margin: 0;
}
.button.gradient {
    background-image: -moz-linear-gradient(center bottom , rgba(0, 0, 0, 0.3) 1%, rgba(255, 255, 255, 0.2) 100%);
}
.button.gradient:hover {
    background-image: -moz-linear-gradient(center bottom , rgba(0, 0, 0, 0.45) 1%, rgba(255, 255, 255, 0.3) 100%);
}

```

```

form.register {
    background-color: white;
    height: 450px;
    margin: 80px auto 0;
    margin-bottom: 50px;
    padding: 5px;
    width: 900px;
}
form p{
    clear: both;
    color: gray;
    font-size: 15px;
    margin: 0;
    padding: 4px;
}

```

```

}
form.register fieldset.row1 {
  border-top: 1px solid #f5f5f5;
  float: left;
  padding: 30px;
  width: 770px;
}
form.register fieldset.row1 label {
  float: left;
  margin-right: 6px;
  margin-top: 2px;
  text-align: right;
  width: 140px;
}

```

```

form.register legend {
  color: #abda0f;
  font-size: 14px;
  font-weight: 100;
  margin-left: 14px;
  padding: 2px;
}

```

```

form.register label {
  color: #444;
  float: left;
  margin-right: 0px;
  margin-top: 2px;
  text-align: right;
  width: 98px;
}

```

```

form.register input,select {
  color: #505050;
  float: left;
  margin-right: 5px;
  width: 140px;
}

```

```

form.register input[type="text"] {
  border: 1px solid #e1e1e1;
  height: 18px; width:200px;
}

```

```

form.register input.short {
  color: #505050;
  width: 40px;
}

```

```
form.register input[type="password"] {
  border: 1px solid #e1e1e1;
  height: 18px;
}
```

```
.logo1 h1{ float:left; color:green; font-size:45px;
```

```
margin-top:-70px;
margin-top:-70px;
}
```

```
.add-button{
  background-color: #bc1815;
  border: medium none;
  border-radius: 3px;
  box-shadow: 0 -3px 1px rgba(0, 0, 0, 0.45) inset, 0 2px 2px rgba(0, 0, 0, 0.25);
  color: white;
  margin-right: 25px;
  margin-top:30px;
  padding: 8px 14px 10px;

  text-shadow: 1px 1px 0 rgba(0, 0, 0, 0.5);
}
```

```
.table{

margin-top:60px; font-weight:solid;
}
.logo2 h1{
  font-size:45px;
  margin-top:30px; color:green;
}
```

```
html .login-wrap {
  margin-top: 40px;
  margin-left: 450px;
  position: relative;
  background: #ecf0f1;
  width: 350px;
  border-radius: 5px;
  box-shadow: 3px 3px 10px #333;
  padding: 15px;
}
html .login-wrap h2 {
  text-align: center;
  font-weight: 200;
```

```

    font-size: 2em;
    margin-top: 10px;
    color: #34495e;
}
html .login-wrap .form {
    padding-top: 20px;
}
html .login-wrap .form input[type="text"],
html .login-wrap .form input[type="password"],
html .login-wrap .form select,
html .login-wrap .form button {
    width: 80%;
    margin-left: 10%;
    margin-bottom: 25px;
    height: 40px;
    border-radius: 5px;
    outline: 0;
    -moz-outline-style: none;
}
html .login-wrap .form input[type="text"],
html .login-wrap .form input[type="password"] {
    border: 1px solid #bbb;
    padding: 0 0 0 10px;
    font-size: 14px;
}
html .login-wrap .form input[type="text"]:focus,
html .login-wrap .form input[type="password"]:focus {
    border: 1px solid #3498db;
}
html .login-wrap .form a {
    text-align: center;
    font-size: 10px;
    color: #3498db;
}
html .login-wrap .form a p {
    padding-bottom: 10px;
}
html .login-wrap .form button {
    background: #e74c3c;
    border: none;
    color: white;
    font-size: 18px;
    font-weight: 200;
    cursor: pointer;
    transition: box-shadow .4s ease;
}
html .login-wrap .form button:hover {
    box-shadow: 1px 1px 5px #555;
}
html .login-wrap .form button:active {

```

```
    box-shadow: 1px 1px 7px #222;
}
html .login-wrap:after {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    height: 5px;
    border-radius: 5px 5px 0 0;
}
```

```
.col1{

    border-right: 1px solid #f2f2f2;

    margin-top: 30px;
    margin-left: 30px;
    width:600px;

}
```

```
.col1 p{

    color: black;
    margin-left: 10px;
    margin-top: 30px;
    padding-bottom: 20px;
    line-height: 20px;
}
```

```
.col1 h1{
    margin-left: 10px;
    margin-top: 100px;
}
```

```
.col1 h2{
    margin-top: 50px;
    margin-bottom: -20px;
}
```

```
.search{

    margin-top: 0px;
    margin-right: 30px;
```

```
float: right;
}
.search ul li{list-style:none; float:left; margin-right:10px;}
.text-filed{ width:230px; height:35px; background:#f0eceb; line-height:35px; text-indent:5px;
border:none;}
```

```
table th{

    text-align:center;
    background: #b0cc7f;
    border: 0px;
}
```

```
table tr,td{
background:#d7e1c5;
border: 0px;
text-align:left;
}
```

```
.right img{
    margin-top: 40px;
    margin-left: 500px;
    margin-bottom: 10px;

}
```

```
.right h1{

    text-align:center;
    color:green;
    font-size: 55px;
    font-style: bold;
}
```


CHAPTER – 6

TESTING

Methodology used for testing

The completion of a system will be achieved only after it has been thoroughly tested. Though this gives a feel the project is completed, there cannot be any project without going through this stage. Hence in this stage it is decided whether the project can undergo the real time environment execution without any break downs, therefore a package can be rejected even at this stage.

Testing methods

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

- 1) **Black box testing** - Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.
- 2) **White box testing** - White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these). White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.
- 3) **Grey Box Testing** - Grey box testing involves having access to internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black-box level. Manipulating input data and formatting output do not qualify as "grey box," because the input and output are clearly outside of the "black-box" that we are calling the system under test. This distinction is particularly important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for test. Grey box testing may also include reverse engineering to determine, for instance, boundary values or error messages.

- 4) **Acceptance testing** - Acceptance testing can mean one of two things:
1. A smoke test is used as an acceptance test prior to introducing a build to the main testing process.
 2. Acceptance testing performed by the customer is known as user acceptance testing (UAT).
- 5) **Regression Testing** - Regression testing is any type of software testing that seeks to uncover software regressions. Such regression occurs whenever software functionality that was previously working correctly stops working as intended. Typically regressions occur as an unintended consequence of program changes. Common methods of regression testing include re-running previously run tests and checking whether previously fixed faults have re-emerged.
- 6) **Non Functional Software Testing - Special** methods exist to test non-functional aspects of software.
- Performance testing checks to see if the software can handle large quantities of data or users. This is generally referred to as software scalability. This activity of Non Functional Software Testing is often times referred to as Load Testing.
 - Stability testing checks to see if the software can continuously function well in or above an acceptable period. This activity of Non Functional Software Testing is often times referred to as indurations test.
 - Usability testing is needed to check if the user interface is easy to use and understand.
 - Security testing is essential for software which processes confidential data and to prevent system intrusion by hackers.
 - Internationalization and localization is needed to test these aspects of software, for which a pseudo localization method can be used.

CHAPTER - 7

CONCLUSION AND

FUTURE WORK

CONCLUSION

The project entitled as **College Admin Management System** is the system that deals with the issues related to a particular institution.

- This project is successfully implemented with all the features mentioned in system requirements specification.
- The application provides appropriate information to users according to the chosen service.
- The project is designed keeping in view the day to day problems faced by a college.
- Deployment of our application will certainly help the college to reduce unnecessary wastage of time in personally going to each department for some information.

Awareness and right information about any college is essential for both the development of student as well as faculty. So this serves the right purpose in achieving the desired requirements of both the communities.

FUTURE ENHANCEMENT:

- Live Lectures can be implemented so that students can attend live lectures from anywhere.
- Scheduling of the staff. i.e. , time table setting of the staff
- Further, the faculty can upload the videos of their lectures on to this site and students who had missed those classes can view those videos.

REFERENCES

Books:

1. Internet & World Wide Web: How to Program Deitel, PJ Deitel.
2. Web Development with Java Server Pages BY Duane K.Fields and Mark A.Kolb.
3. The Complete Reference Java2 HerbertSchildt.
4. Core Servlets and Java Server Pages By Marty Hall.
5. Apache Jakarta-Tomcat by James Goodwill.

Web Sites:

- 1] [Google](#)
- 2] [Stack Overflow](#)
- 3] [Tutorials Point](#)
- 4] [W3schools](#)