COLLEGE ERP SYSTEM

Dada Patil College

MARCH 27, 2021
WEBSOFT IT SOLUTION

ACKNOWLEDGEMENT

I would like to express my sincere and deep gratitude to my internal project guide **Mr Venugopal Narsingoju,** for his valuable guidance and suggestions. I would also like to thank the coordinator for giving me guidance and timely help.

Lastly I thanks all those who have directly or indirectly motivated and helped us to complete this project.

Ms. ANERI SHAH.

(MCA - SEM VI)

WEB SOFT IT SOLUTION PUNE

Office Address: 1st Floor, Office No -2, Building C-2, Manikmoti Complex, Near Jivandhara Hospital,
Pune- Satara Road, Katraj Chowk, Pune- 411046.
Mobile No: +91-7755945544 E-mail: websoftitsolutionpune@gmail.com
Website: www. websoftitsolution.com

REF: Int-WS209556-2020

INTERNSHIP LETTER

This is to certify that **Ms. Aneri Shah** student of MCA from **"M.E.S. Institute of Management and Career Course, Pune"** has successfully delivered Internship on **"School ERP System"** as a partial fulfillment of requirement towards of her MCA project.

University Name- Savitribai Phule Pune University, Pune.

Duration- 1st November 2020 to 30 April 2021

As abided by intellectual property and confidentiality policy of WEBSOFT IT SOLUTION Pune.

We wish her every success in life.

AUTHORIZED PERSON SIGN

WEB SOFT IT SOLUTION PUNE

www.websoftitsolution.com

Index

CHAPTER 1
INTRODUCTION

1.1 COMPANY PROFILE

Web Soft IT Solution is a Software Development. Also, leading solution providers in Information Technology Training and Corporate customers. With our headquarters in Pune, one of the IT hubs of India, our training activities are spread over many locations in the state of Maharashtra. The Corporate training programme offered by Web Soft IT Solution companies to improve the efficiency level of their Students by enhancing their Technical and soft skills. Programs tend to sharpening of theoretical concepts and phenomenon and focus more on its practical application. In addition to the lectures, most programs make use of real case studies/discussion intensively. All programs tend to provide ready-to-use skills at your work place. We Provides Training to Engineering Student and Working Employees who recover the knowledge gap between Industry and Academic level through Experts Trainer. All Trainer belongs to IT Industry and MNC Company having the 7+ Experience.

OUR MISSION

To provide training to candidates in various technologies by experts who have real-time industry knowledge so that candidates can start their career and be in sync with the industry.

WHY WSTs

Websoft IT Solution's highly qualified talent pool of technical graduates is one of the largest in the India and Recover the Knowledge Gap between Industry and Academic Education. Trained You Not for Only Placement but also for Technical Knowledge and as a student you can acquire technical skills if you feel that is something appealing to you, and would be important for you.

OUR SERVICES

Web Soft IT Solution provides various Software Development services to the clients located worldwide in order to rationalize their business processes and e-enabling their business. We, at Web Soft IT Solution strongly believe that technology is a true business enhancer and you should not implement technology for the sake of it. That's why; we help you make best use of information technology. With the use of our services such as application development, application migration and application maintenance for your existing applications, you can formulate the best possible use of technology. We are a team of professional organization teamed by competent, committed, qualified and experienced personnel in various field. With the help of our commitment to professionalism and excellence. The programmers we design are developed to meet specific organizational needs. We provide a service that provides clients with value for each rupee invested. So feel free to come forward and avail the opportunity of getting reasonably priced consultancy services from us.

IT PROVIDES THE FOLLOWING SERVICES TO THE CLIENTS:-
Technology Services-

- Data Warehousing
- Data Migration
- High Availability
- Internet of Things
- Java Technology
- Linux and Unix

- Architecture
- Windows and Dot Net
- Virtualization

Business Services-

- Business Analytics
- Business Process Services
- Customer Experience
- Customer Relationship Management
- EntStudent Management systemrise Content Management
- EntStudent Management systemrise Management
- Management & Retail Services
- Database and IT Infrastructure
- Project Portfolio Management

1.2. Existing System and Need of the System

College ERP System is a System that manages the records of student regarding admission, courses, examination part, and exam result summary for particular student and also other facilities to provide generating reports for faculty.

The system is designed to help for keeping data, storing, manipulating data and analyzing the data. Extensive information is available at your fingertips through this System. Viewing student data, managing student information and category and for examination, courses management, scheduling exam, result and related issues are made simple and easy. There are custom search capabilities to aid in finding student information and working on student records. This can make the system easier to navigate and to use maximizing the effectiveness of time and other resources. COLLEGE ERP SYSTEM allows the keeping of personnel data in a form that can be easily accessed and analyzed in a consistent way.

Each of modules in COLLEGE ERP SYSTEM is covering many other student aspects from application to retirement. The system records basic personal information, admission information, courses information regarding student. Leading edge systems provide the ability to "read" applications and enter relevant data to applicable database fields, notify student and provide result.

COLLEGE ERP SYSTEM function involves:

- Manage new admissions
- Manage courses
- Manage Events
- Student Basic Information
- Manage faculty
- Exam scheduling
- Result management
- Declaration global notes
- Manage attendance
- Manage batches
- Report generation

In COLLEGE ERP SYSTEM, every user has a Login ID and Password. Also all the users have different permission rights to access the applications. These rights are Dynamic and can be changed.

There are three main roles in the system. Admin, Faculty and Student. Admin has complete access to the whole system, while faculty is concerned with courses, check their tasks, checking examination results for the student. Students role are responsible for the use of the system.

The Admin role can be as follow:

- Add/Modify faculty
- Add/Modify student
- Add/Modify batches
- Attendance
- Search student(course, progress, batch, , basic information)
- Search faculty(course, batch, total leave, basic information)
- Reports
- Add/Modify Events

The faculty role can be as follow:

- Manage task(upcoming event, lecture flow, assignment, self-assessment)
- Search student(course, batch student status, progress)
- Reporting(student information, daily report)

The student role can be as follow:

- Viewing profile
- Assessment(exam detail, assignment, lecture flow)
- Supporting document
- Suggestion

Need

The project is about to handle all the information of the student regarding course and examination. Also it manages resources which were managed and handled by manpower

previously. The main purpose of the project is to integrate distinct sections of the organization into consistent manner so that complex functions can be handled smoothly by any technical or non-technical persons.

The project aims at the following matters:

- To manage information of student, faculty and courses.
- Consistently update information of all the students.
- Reports.
- Assistance in decision-making.

1.3 Scope of Work

- Different people, place from different departments can view the same information about Student Information.
- To enable the head and technical supporting group to access the system from anywhere.
- To enable the Student to view as well as raise suggestion from anywhere.
- To enable the student evolution with giving online exam and get the result on the spot.

1.4. Hardware and Software Requirement Study Hardware Specifications (Client Side)

RAM	Minimum 2GB and above
Hard Disk	Minimum 80 GB and above
Processor	Dual Core and above

Hardware Specifications (Server Side)

RAM	Minimum 4GB and above
Hard Disk	Minimum 250 GB and above
Processor	Core 2 Duo and above

Software Specifications (Client Side)

Operating System	Windows XP/Later
Web Browser	Internet Explorer-6/Later Mozilla Firefox

Software Specifications (Server Side)

Operating System	Windows XP/Later
Web Browser	Internet Explorer-6/Later Mozilla Firefox
Technology	Asp.Net,C#.Net
Database	Microsoft SQL Server 2008(R2)

Development Tool(Editor)	Microsoft Visual Studio 2010(Ultimate)
Supporting Technology	Ajax, JavaScript, JQuery, Html5, Css3

1.5. Detail Description of Technology Used

- **ASP.NET**

ASP.NET is a set of web development technologies marked be Microsoft. Programmers can use it to build dynamic web sites. Web applications and XML web services. It is part of Microsoft's .net platform and is the successor to Microsoft's Active Server Pages (ASP) technology.

➢ **Principles of ASP.NET**

Even though ASP.NET takes its name from Microsoft's old web development technology, ASP, the two differ significantly. Microsoft has completely rebuilt ASP.NET, based on the Common Language Runtime (CLR) shared by all Microsoft .NET applications. Programmers can write ASP.NET code using any of the different programming languages supported by the .net framework, usually (proprietary) Visual Basic .NET, Jscript .NET, or (Standardized) C#, but also including open-source languages such as Perl and Python. ASP.NET has performance benefits over previous script-based technologies because the server-side code is compiled to one or a few DLL files on a web server. ASP.NET attempts to simplify developer's transition from Windows application development to web development by allowing 15 them to build pages composed to controls similar to a Windows user interface. A web control, such as a button or label, functions in very much the same way as events. Controls know how to render themselves: whereas windows controls draw themselves to the screen, web controls produce segments of HTML which form part of the resulting page sent to the end-user's browser.

ASP.NET encourages the programmer to develop applications using an event-driven GUI paradigm, rather than in the conventional web scripting fashion. The framework attempts to combine existing technologies such as JAVA Scripts with internal components like —View State‖ to bring persistent (inter-request) state to the inherently stateless web environment.

➢ **C Sharp**

In June 2000, Microsoft announced both the .NET platform and a new programming language called C#. C# is a strongly-typed object oriented language designed to give the optimum blend of simplicity, expressiveness, and performance. The .NET platform is centered on a Common Language Runtime (similar to JVM) and a set of libraries which can be exploited by a wide

variety of languages which are able to work together by all compiling to an intermediate language (IL). C# and well with .NET, and some features of .NET are there to work well with C# (though .NET aims to work well with many languages). This article is mostly concerned with C#, but sometimes it is useful to discuss .NET too. The C# language was built with the hindsight of many languages, but most notably Java and C++. It was co-authored by Anders Hejlsberg (who is famous for the design of the Delphi language), and Scott Willamette.

> **New Features in C#**

The C# language is standardized through ECMA International. This allows for a standardization of the language without the fear of one company controlling it. Said another way, C# is a non-proprietary language that no single company can control. The ECMA committee has recently accepted two new specifications. These are the Standard EMCA-334: C# language Specification and the standard EMCA-335: Common Language Infrastructure (CLI) Specification. I took a quick look at these. The acceptance of EMCA-334 is the approval of the third edition of the C# language. This standard specifies the syntax and constraints used to create a standard C# program. Additionally, it defines the semantic rules for interpreting C# programs and the restrictions and limits that must be met to be considered a conforming version of C#. More specifically, what does this mean to a C# 17 developer? It means that a number of new features can be used with confidence because future C# compilers will be supporting them. These features include:

- Pragma directives
- Anonymous methods
- Static classes
- Partial declarations
- Property get and set assessors
- Generic types and methods
- Iterators
- Null-able types
- With the changes that have been made to the C# language, it is also not surprising that changes also needed to be made within the standardized CLI specification. Changes in the CLI based on ECMA-335 are:
- Support for generic types and methods.
- Additional prefixes of constrained. No. and read-only.

- Added a number of members and types to the Type library.

➢ Made changes to the specification including the addition of a ―Debug Interchange Format section. In the CLI, there are numerous types and members that were added to the type library. These include a number of generics in the System 18 namespace ranging from Action <T> to Typed Reference. The System Collections Generics namespace was also added along with numerous types within it. The system reflection namespace was updated to allow for support of generics reflection as well to include new enumeration values to support non-standard encoding of string formatting attributes. The System Runtime Compiler Services namespace was updated to include types Compilation Relaxations and Compilation Relaxations Attribute. Finally, the System threading Parallel namespace now contains a number of new types for supporting multi-threaded CPUs. As you can see, C# and the .NET Framework are evolving. These new additions help to boost C# into being an even more powerful, non-proprietary, standardized language. Although Microsoft may be initiating the changes, it is good to know that a standards board is controlling the language. In my opinion, this gives C# developers some added protection as the C# language continues to evolve.

DOT NET FRAMEWORK:

The **Microsoft .NET Framework** is a software framework that can be installed on computers running Microsoft Windows operating systems. It includes a large library of coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The 19 .NET Framework is a key Microsoft offering and is intended to be used by most new applications created for the Windows platform.

The framework's Base Class Library provides a large range of features including user interface, data and data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. The class library is used by programmers, who combine it with their own code to produce applications.

Programs written for the .NET Framework execute in a software environment that manages the program's runtime requirements. Also part of the .NET Framework, this runtime environment is known as the Common Language Runtime (CLR). The CLR provides the appearance of an application virtual machine so that programmers need not consider the capabilities of the specific CPU that will execute the program. The CLR also provides other important services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework.

Version 3.0 of the .NET Framework is included with Windows Server 2008 and Windows Vista. The current version of the framework can also be installed on Windows XP and the Windows Server 2003 20 family of operating systems.[2] A reduced version of the .NET Framework is also available on Windows Mobile platforms, including smart phones as the .NET Compact Framework. Version 4.0 of the framework was released as a public Beta on 20 May 2009.

Common Language Runtime Engine

The Common Language Runtime (CLR) is the virtual machine component of the .NET framework. All .NET programs execute under the supervision of the CLR, guaranteeing certain properties and behaviors in the areas of memory management, security, and exception handling.

➢ **Language Independence**

The .NET Framework introduces a Common Type System, or CTS. The CTS specification defines all possible data types and programming constructs supported by the CLR and how they may or may not interact with each other. Because of this feature, the .NET Framework supports the exchange of instances of types between programs written in any of the .NET languages. This is discussed in more detail in Microsoft .NET Languages.

➢ **Base Class Library**

The Base Class Library (BCL), part of the Framework Class Library (FCL), is a library of functionality available to all languages 21 using the .NET Framework. The BCL provides classes which encapsulate a number of common functions, including file reading and writing, graphic rendering, database interaction and XML document manipulation.

➢ **Simplified Deployment**

The .NET framework includes design features and tools that help manage the installation of computer software to ensure that it does not interfere with previously installed software, and that it conforms to security requirements.

➢ **Security**

The design is meant to address some of the vulnerabilities, such as buffer overflows, that have been exploited by malicious software. Additionally, .NET provides a common security model for all applications.

➢ **Portability**

The design of the .NET Framework allows it to theoretically be platform agnostic, and thus cross-platform compatible. That is, a program written to use the framework should run without change on any type of system for which the framework is implemented. Microsoft's commercial implementations of the framework cover Windows, Windows CE, and the Xbox 360.[4] In addition, Microsoft 22 submits the specifications for the Common Language Infrastructure (which includes the core class libraries, Common Type System, and the Common Intermediate Language),[5][6][7] the C# language,[8] and the C++/CLI language[9] to both ECMA and the ISO, making them available as open standards. This makes it possible for third parties to create compatible implementations of the framework and its languages on other platforms.

➢ **Features of .NET Framework:**

The .NET framework offers powerful capabilities for software development and deployment including independence from a specific language and platform. The .NET platform increases programmer's productivity and decreases the development time.

The .NET framework was designed with three goals in mind:

1. To make Windows application much more reliable, while also providing an application with greater degree of security.

2. To simplify the development of Web application and services that not only works in traditional sense, but on mobile devices as well.

3. To provide a single set of libraries that would work with multiple languages.

- **SQL SERVER 2005 as Back End:** 23 Microsoft SQL Server 2005 is comprehensive, integrated data management and analysis software that enables organizations to reliably manage mission-critical information and confidently run today's increasingly complex business applications. SQL Server 2005 allows companies to gain greater insight.

- **Why SQL Server 2005?**

SQL Server 2005 is a comprehensive database platform providing enterprise-class data management with integrated business intelligence (BI) tools. The SQL Server 2005 database engine provides more secure, reliable storage for both relational and structured data, enabling you to build and manage highly available, performing data applications that you and your people can use to take your business to the next level.

The SQL Server 2005 data engine lies at the core of this enterprise data management solution. Additionally, SQL Server 2005 combines the best in analysis, reporting, integration, and notification. This enables your team to build and deploy cost-effective BI solutions with which they can drive data into every corner of your business through scorecards, dashboards, Web services, and mobile devices.

Close integration with Microsoft Visual Studio, the Microsoft Office System, and a suite of new development tools, including the Business 24 Intelligence Development Studio, sets SQL Server 2005 apart. Whether you are a developer, database administrator, information worker,

or decision maker, SQL Server 2005 provides innovative solutions that help you gain more value from your data.

The following diagram illustrates the core components in SQL Server 2005, showing how SQL Server 2005 is a key part of the Windows Server System in integrating with the Microsoft Windows platform—including the Microsoft Office System and Visual Studio—to offer solutions that deliver data to every corner of your organization.

CHAPTER 2

2.1. Proposed System

The main objective behind the COLLEGE ERP SYSTEM is creating a central system while in traditional approach is to do a manual work in paper. So for that it's very difficult to manage all the record and also difficult to analyze all the record in any departments. A manual work are to lengthy and its very time consuming for the entire department. So for that needs of central system is arising which gives the effective and efficient result within a few time. All departments can access the data with the system and also they can perform a desire task. With that all the data can easily manipulate and get easily whenever anybody wants.

With an COLLEGE ERP SYSTEM in place, people in different departments can view the same information about Student Information, making sure that every single interaction with student is based on accurate information. It also means that manual processes can be automated to free up people's time and make internal processes more efficient. It's all about to how to manage the student record in a master file as well as how admin and faculty can manage the profile and keep information about the student. All the information regarding student can see and edit if necessary apart from that faculty can keep record of student and they can manage information like student profile, exam, reporting or other information.

COLLEGE ERP SYSTEM function involves:

- Manage new admissions
- Manage courses
- Student Basic Information
- Manage faculty
- Exam scheduling
- Result management
- Declaration global note

- Manage Events
- Manage attendance
- Manage batches
- Report generation

The Admin role can be as follow:

- Add/Modify faculty
- Add/Modify student
- Add/Modify batches
- Add/Modify Event
- Attendance
- Search student(course, progress, batch, , basic information)
- Search faculty(course, batch, basic information)
- Reports

The faculty role can be as follow:

- Manage task(upcoming event, lecture flow, assignment, self assessment)
- Search student(course, batch student status, progress)
- Reporting(student information, daily report)

The student role can be as follow:

- Viewing profile
- Assessment(exam detail, assignment, lecture flow)
- Supporting document
- Suggestion

2.2. Objective

- View the all the record of the student and faculty.
- Student can interact with his basic profile which includes course details, personal detail.
- Student can download the appropriate document in the supporting document.
- Student can give the online exam and get the result on the spot time.
- Give the suggestion or complain to the administrator.
- Faculty can get the information about the student, his batches.
- To enable to send a daily report to the administrator.
- Faculty get perform the online assessment of student.
- Keep the information about the student.
- Administrator can add/modify to the student as well as faculty.
- Also can manage the batches and course for particular student.
- Teacher can set the exam paper, and many other tasks
- Admin can able to add/modify the events

2.3. User Requirement

2.3.1 Feasibility Study

The aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the system or not. A feasibility study is carried out from following different aspects:

Technical Feasibility

Technical feasibility corresponds to determination of whether it is technically feasible to develop the software. It refers to the ability of the process to take advantage of the current state of the technology in pursuing further improvement. The technical capability of the available technology should be considered.

The following technical feasibility areas were probed during the feasibility study phase:

- The necessary technology i.e. front-end development tools, back-end database technology for developing the system are already available within the organization.
- The front-end tool proposed in easily compatible with the current hardware configuration in the organization.
- The back-end tool proposed has the capacity to hold the data required for using the new system.
- The System is expandable in many dimensions with respect to addition of more functionality, featured, etc.
- The front-end and back-end technologies provide a way to preserve the accuracy, reliability and ease of access and data security.

Implementation Feasibility

This project can easily be made available online without much consideration of the hardware and software. The only required thing at the applicant's side is the Internet connection and a web browser, which are a no difficult issue these days. A database server and application server

are required to set up at the admin side. After setting up the project online, even the administrator can access the system from anywhere.

Operational Feasibility

The system has been developed for any user who wants to use this system. I have given a demo of my project and the users found the system friendly and easy to use. The interoperability with the existing system is also checked. So they may face certain problems in using the user interface. So keeping this consideration in mind we have provided field for each and every field on the forms. The administrator also may be non-technical, so the user interface is designed in such a way that it gets comfortable for the non-technical person to operate easily.

Fact Finding Methods

To study any system the analyst needs to do collect facts and all relevant information. the facts when expressed in quantitative form are termed as data. The success of any project is depended upon the accuracy of available data. These specific methods for finding information of the system are termed as fact finding techniques. Interview, Questionnaire, Record View and Observations are the different fact finding techniques used by the analyst. The analyst may use more than one technique for investigation.
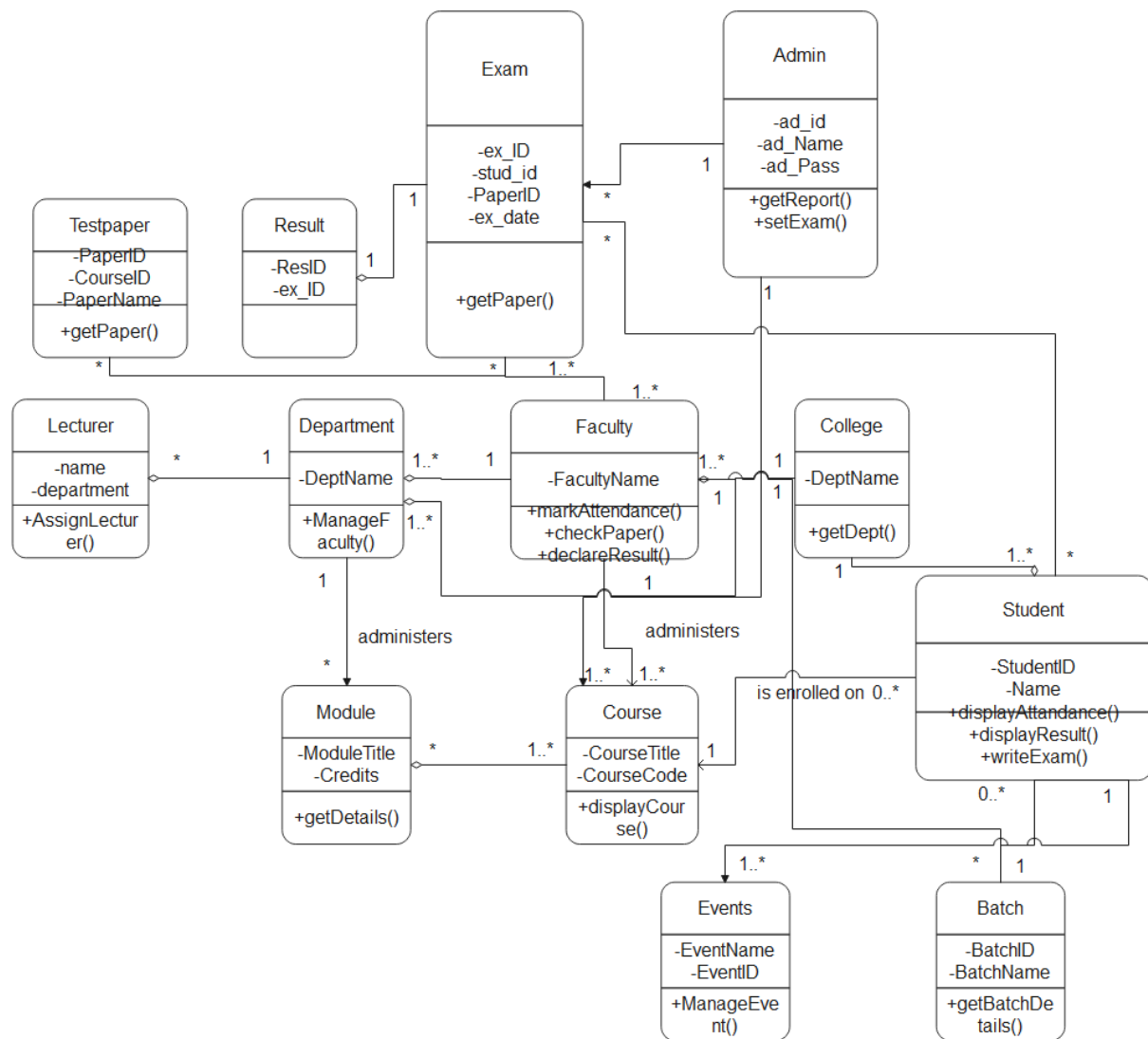
RecordView

 The information related to the system is published in the sources like newspapers, magazines, journals, documents etc. This record review helps the analyst to get valuable information about the system and the organization. During the development of this system we goes through lots of documents and record views and their formats. During the designs of the quotes, invoices and sales orders we see the different type of quotes, invoices and sales orders. Whereas during the report generation of all the quotes and invoices and sales orders we see the record details of various organizations quotes and invoices entries. During the design and working with contact management, lead management, opportunity management and organizations management we goes through the lots of records of different organizations and how they currently uses forms and what are the fields that are very important during the design and working on campaigns , leads, organizations and opportunities.
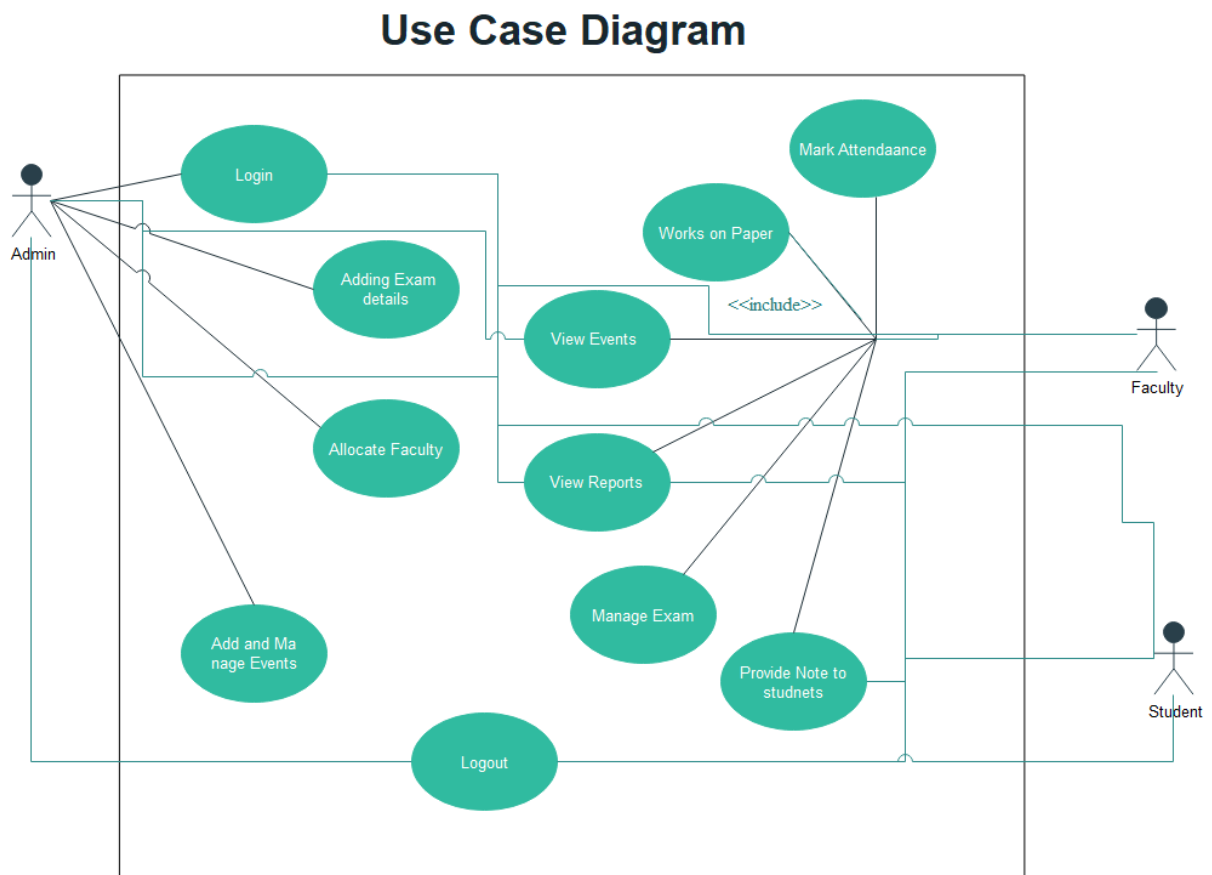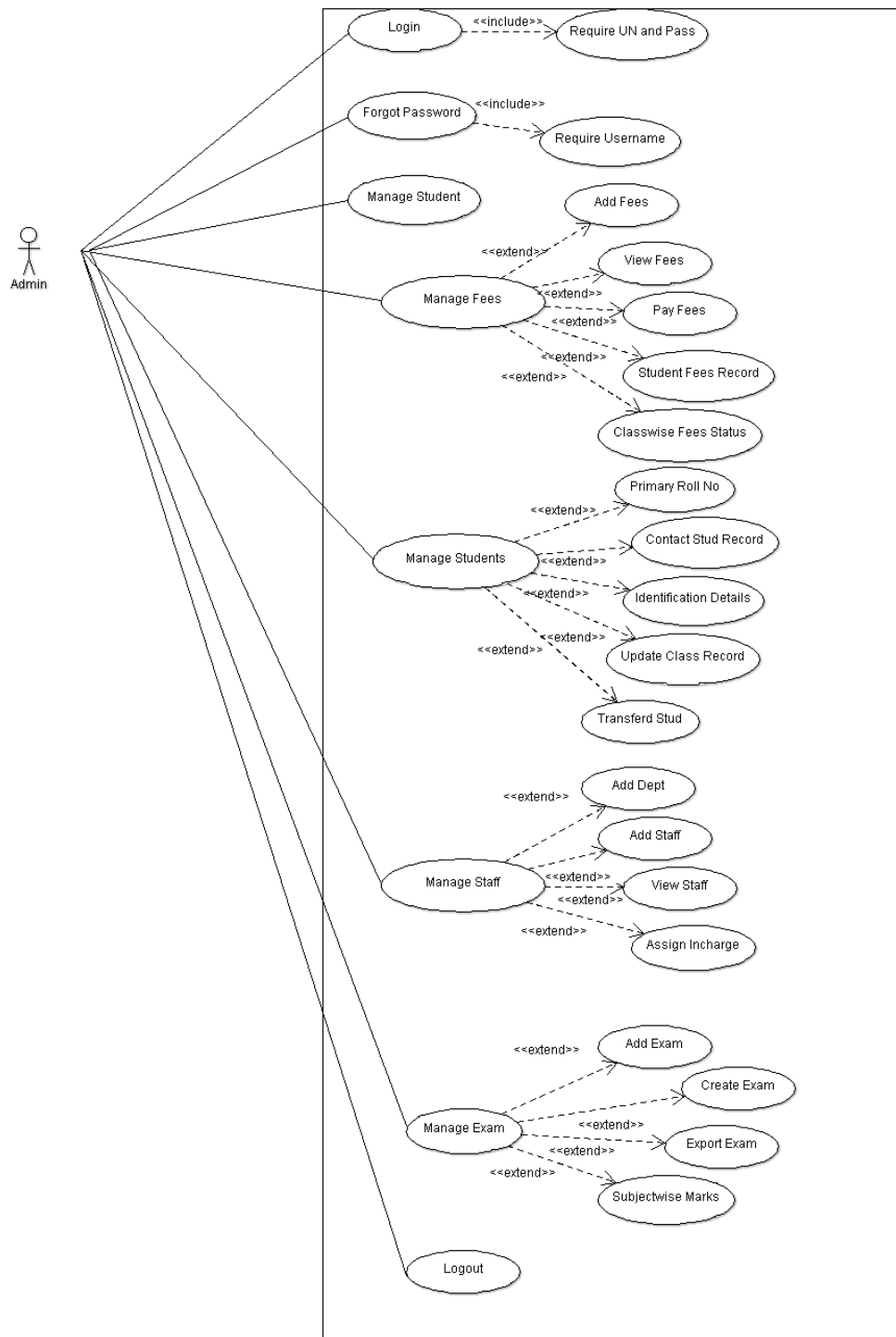
Observation

 Unlike the other fact finding techniques, in this method the analyst himself visits the organization and observes and understand the flow of documents, working of the existing system, the users of the system etc. During the development of this system we observe the work flow of different organizations and how their daily tasks are done. It is observed that how the campaigns, leads and opportunities are related to each other.
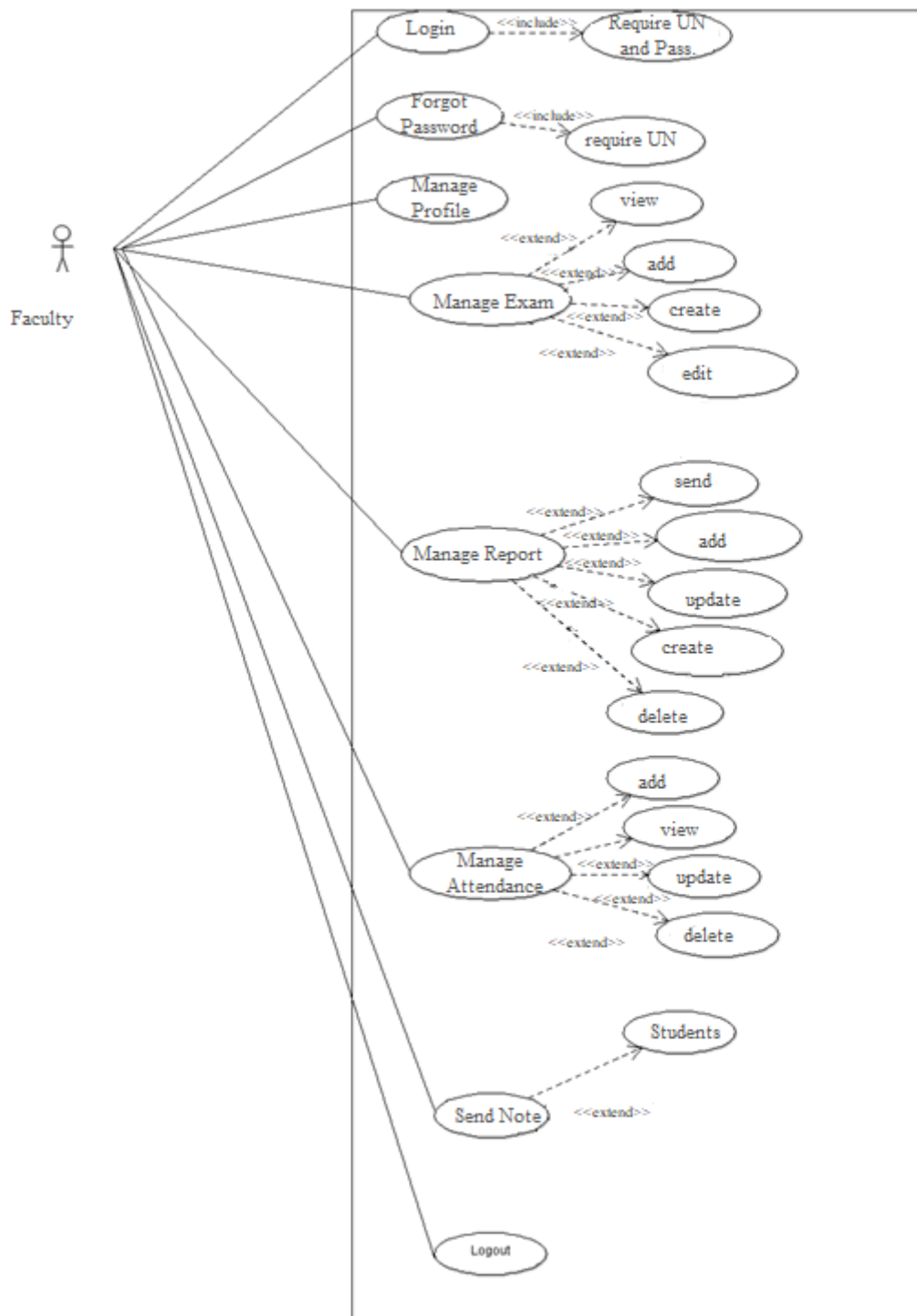
Chapter 3

3.1. Object Diagram

3.2. Class Diagram

3.3. Use Case Diagram

Use Case Diagram

Registration

<<include>> Validate

Login <<extend>>

Forgot password

View Personal Details
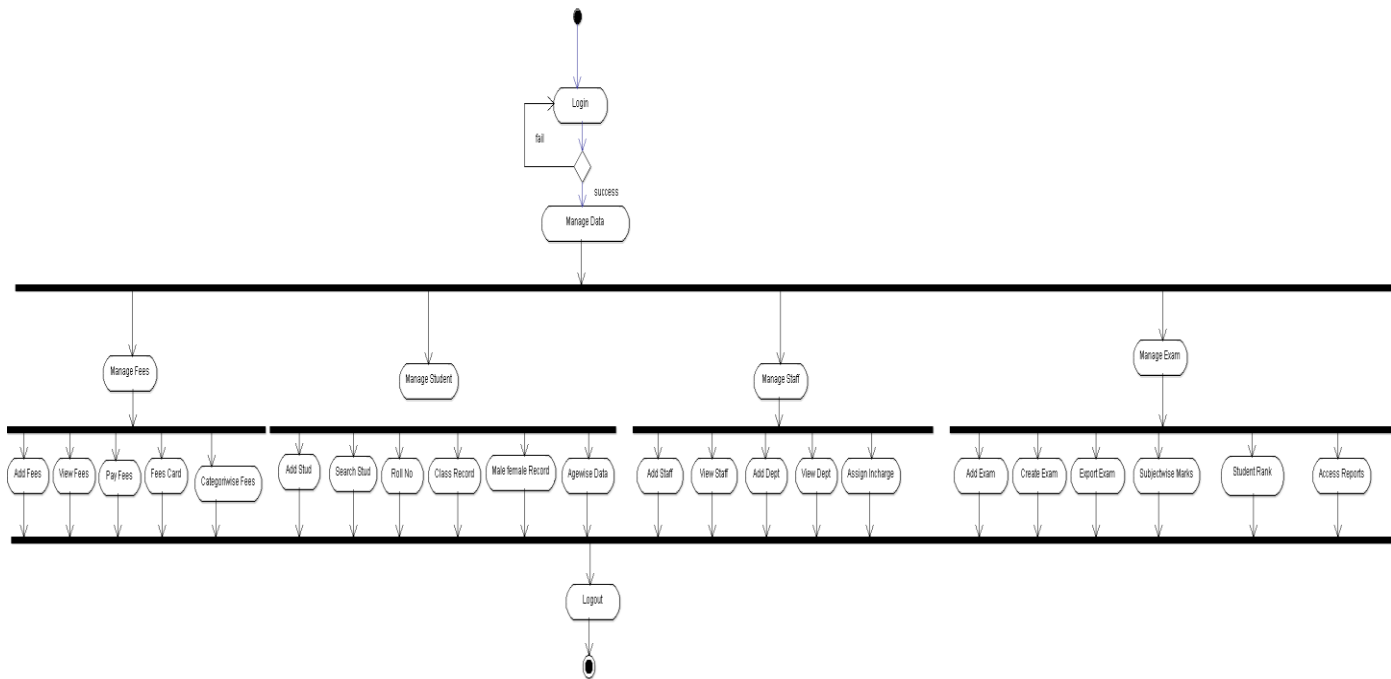
Show Exam Shedule

Show Result

View Exam Details

Check Attendance

View Note

Logout
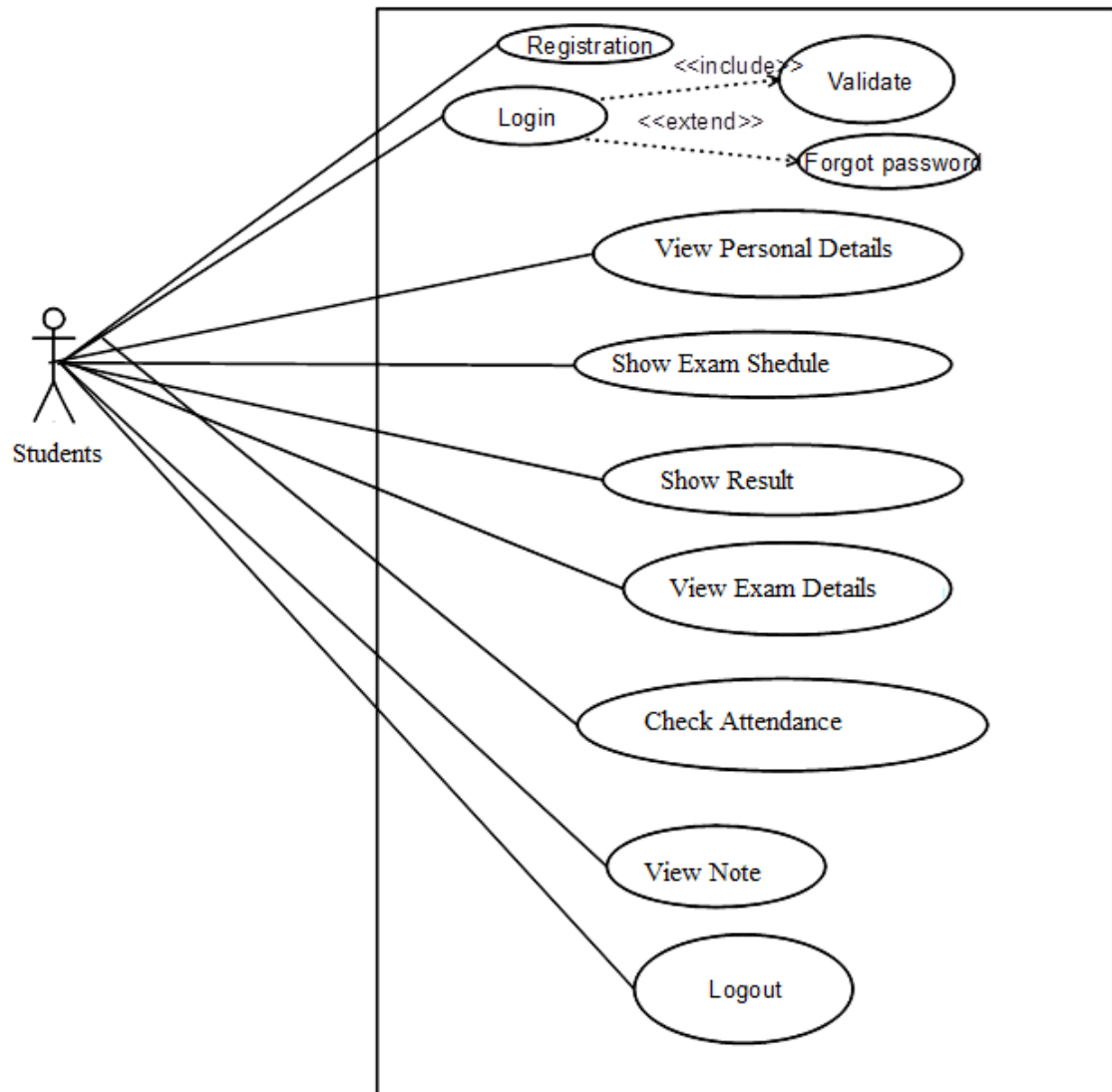
Students

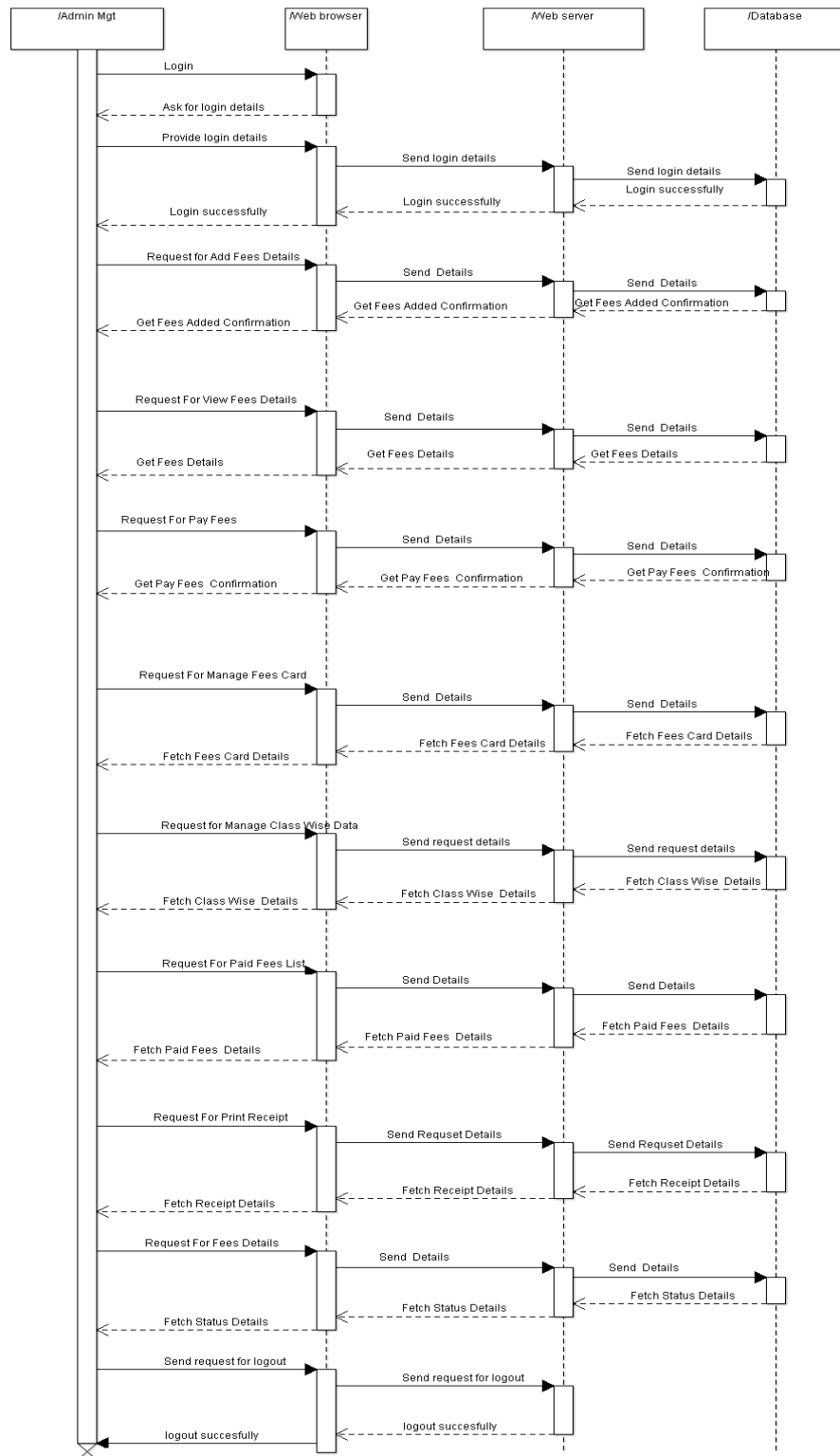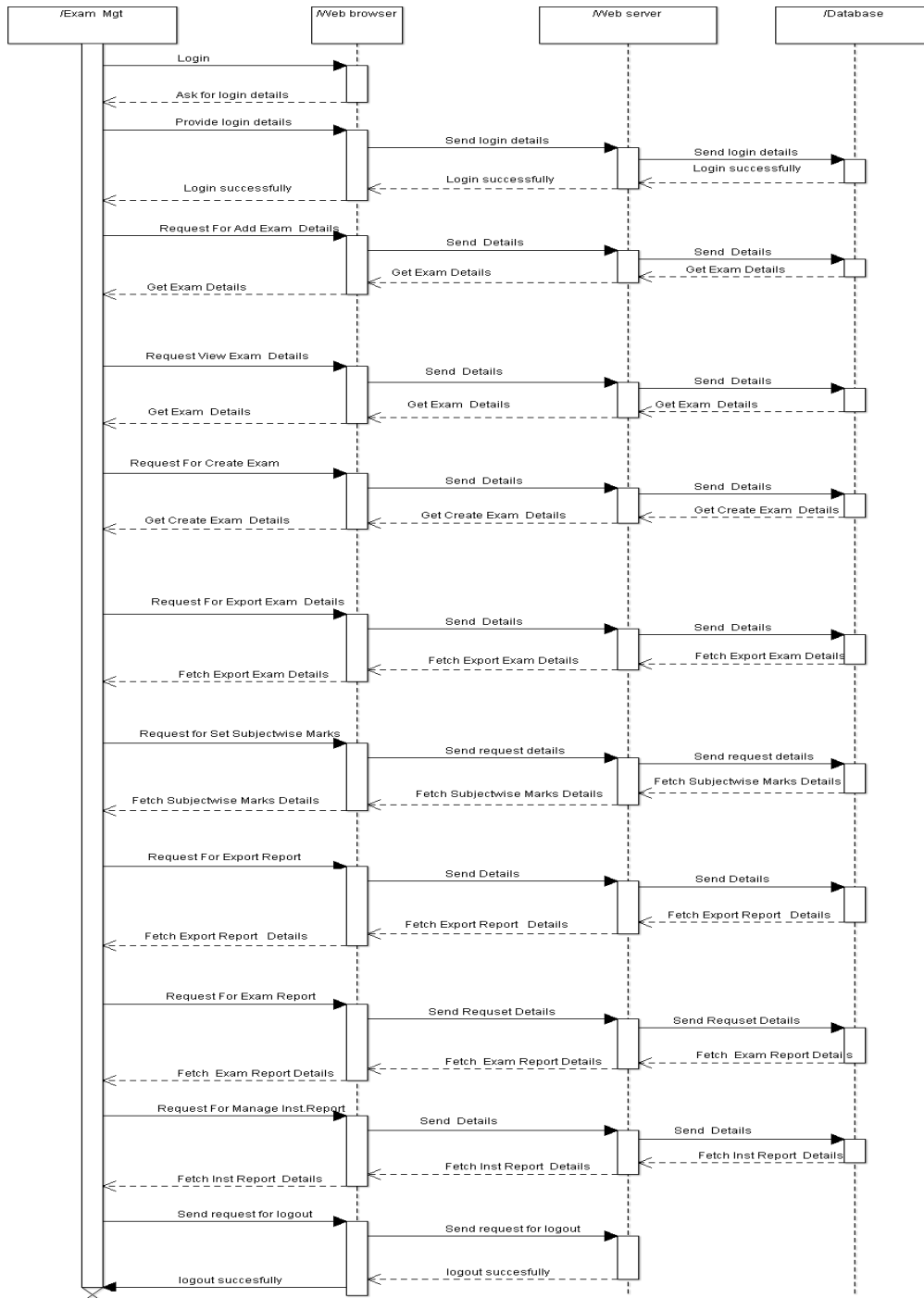# 3.4. Activity Diagram Admin

Global

Admin
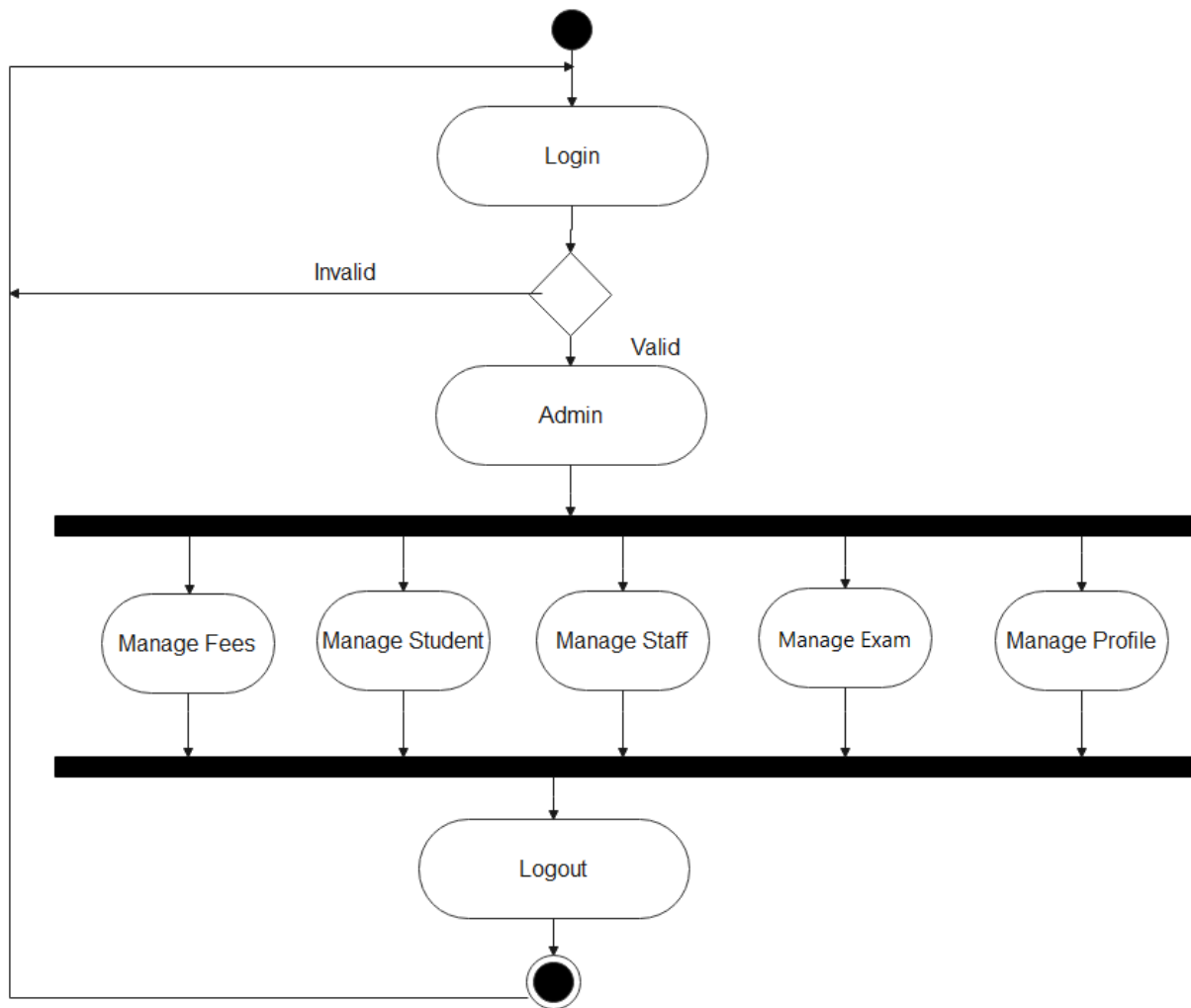
Faculty
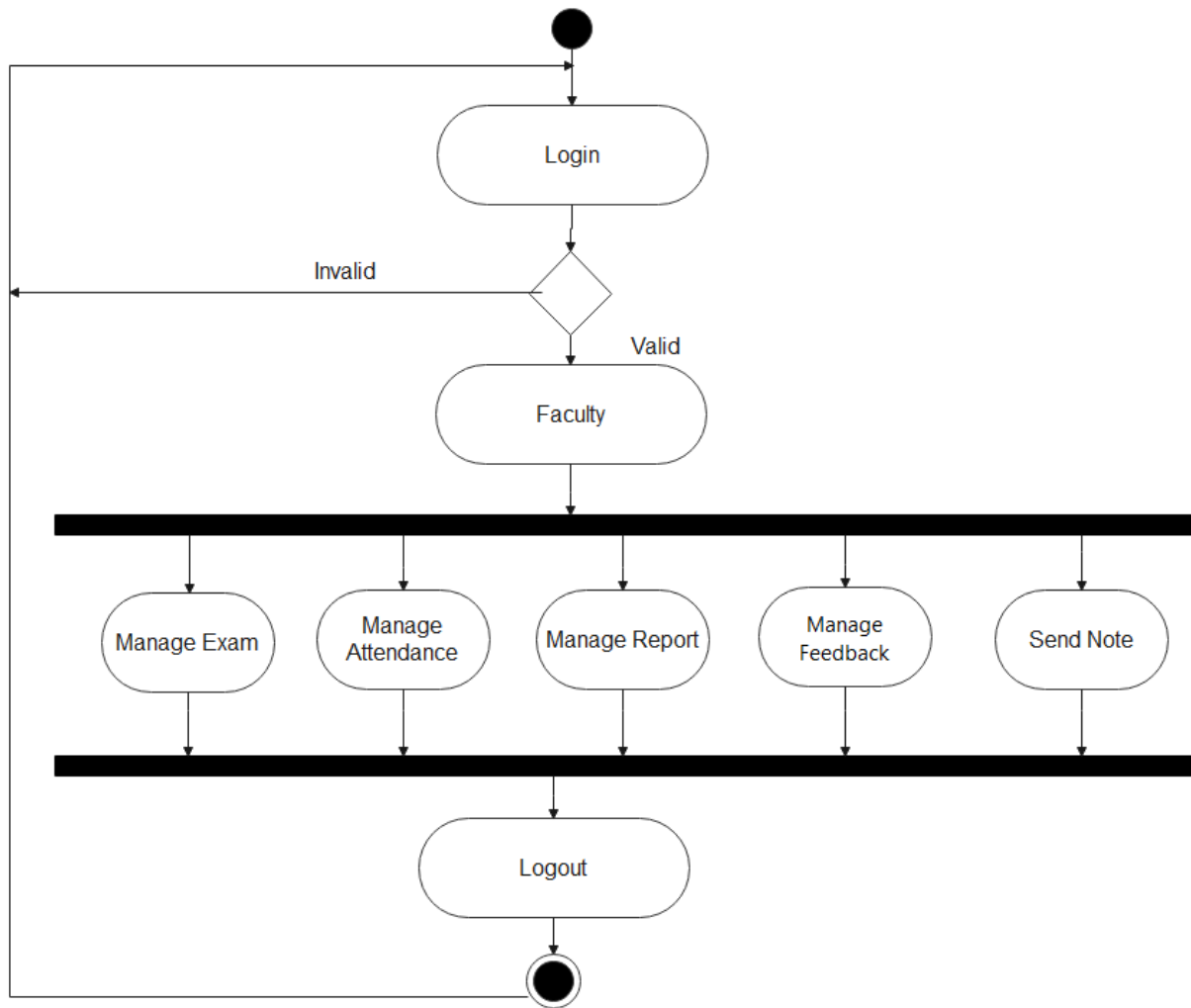
Students
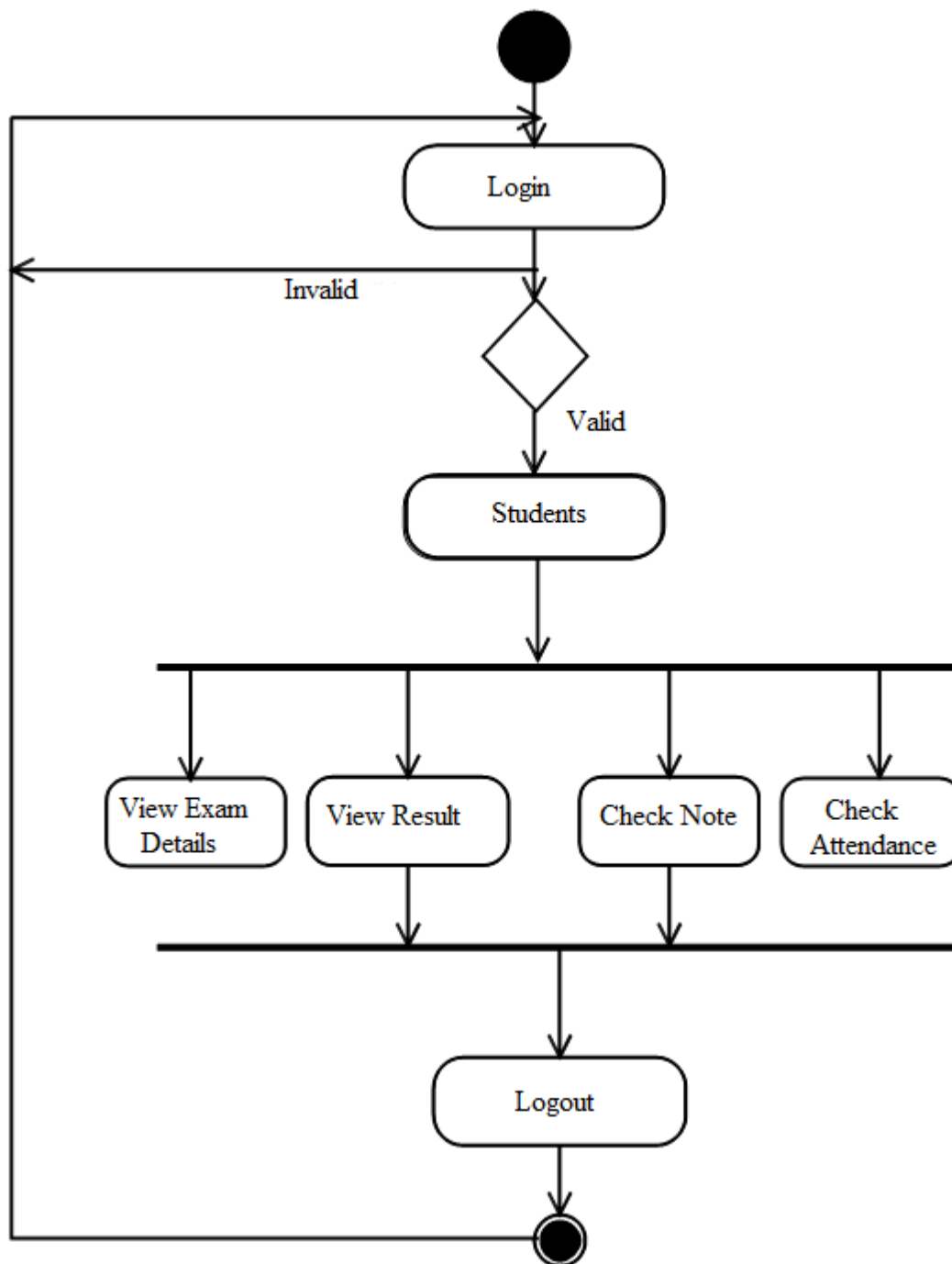
3.5. Sequence Diagram Admin Mgt

Sequence Diagram Exam Mgt

Sequence Diagram Student Mgt

3.6. ER Diagram
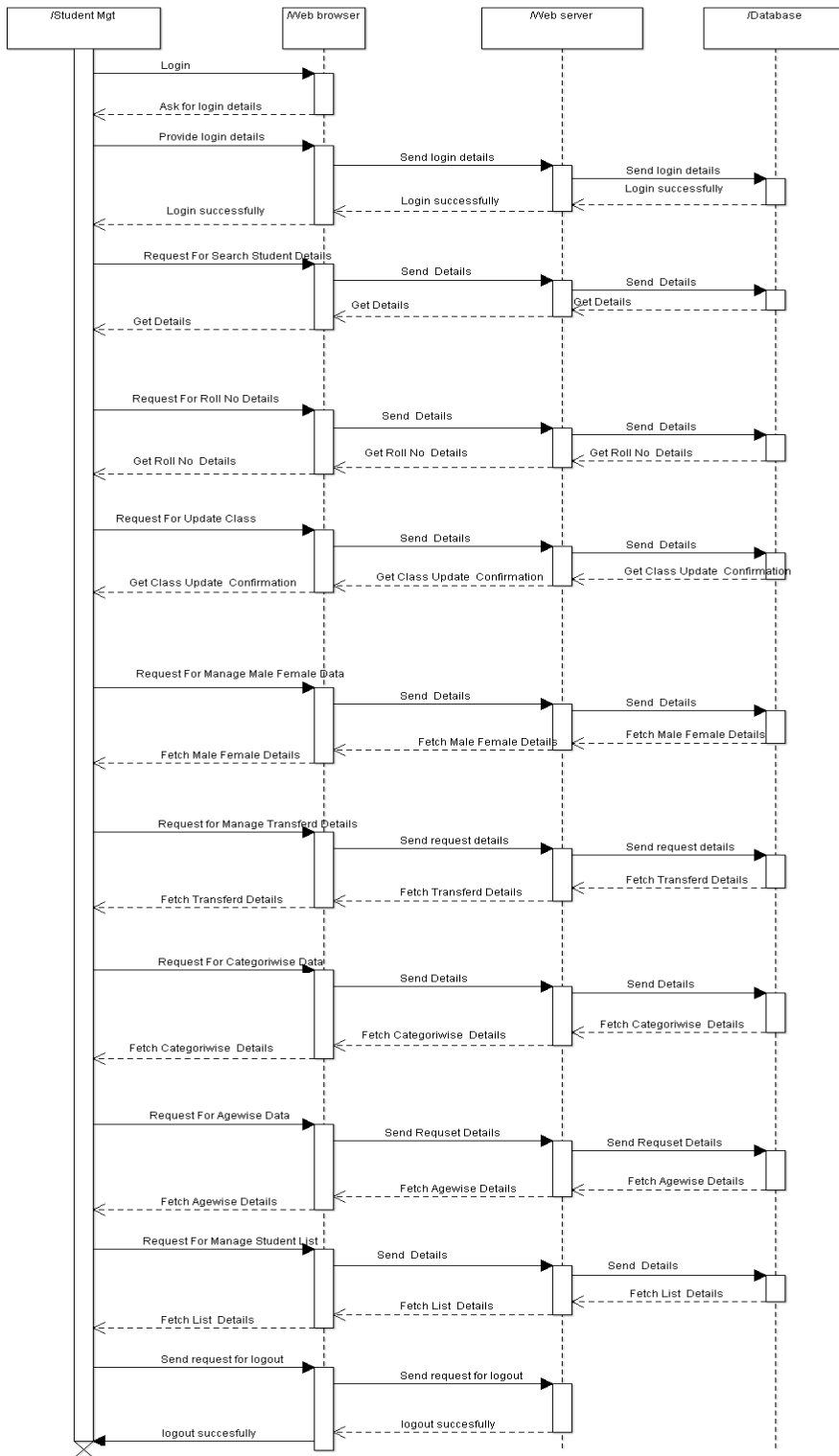
3.7. Module Hierarchy-
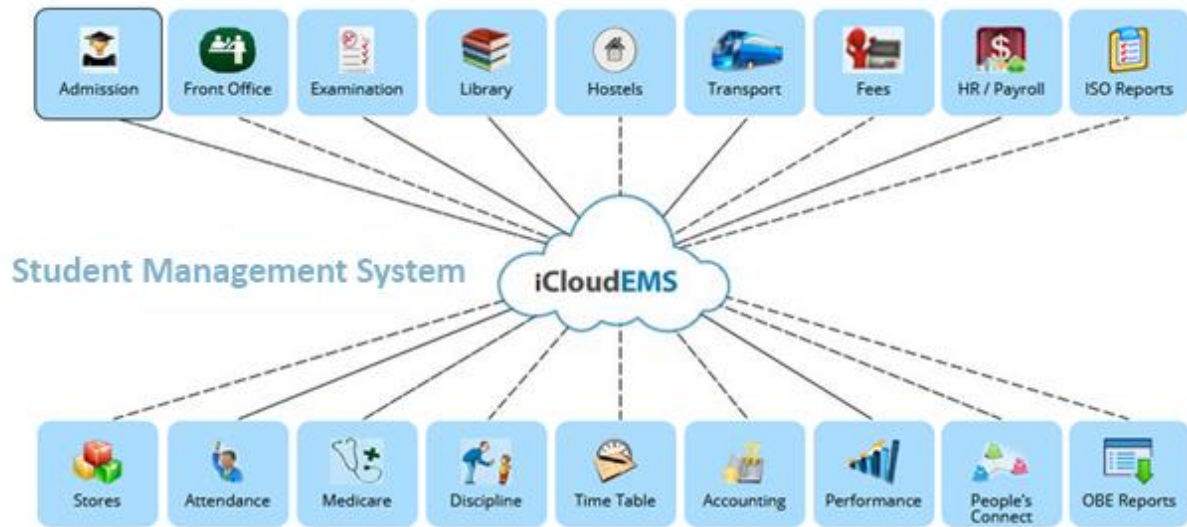
3.8. Component Diagram
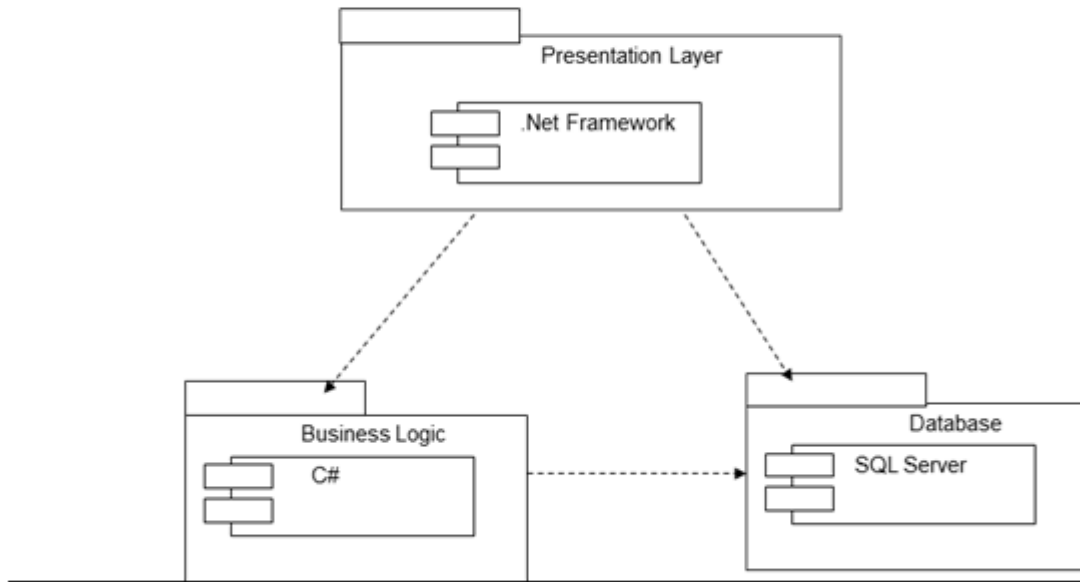
3.9 Deployment Diagram
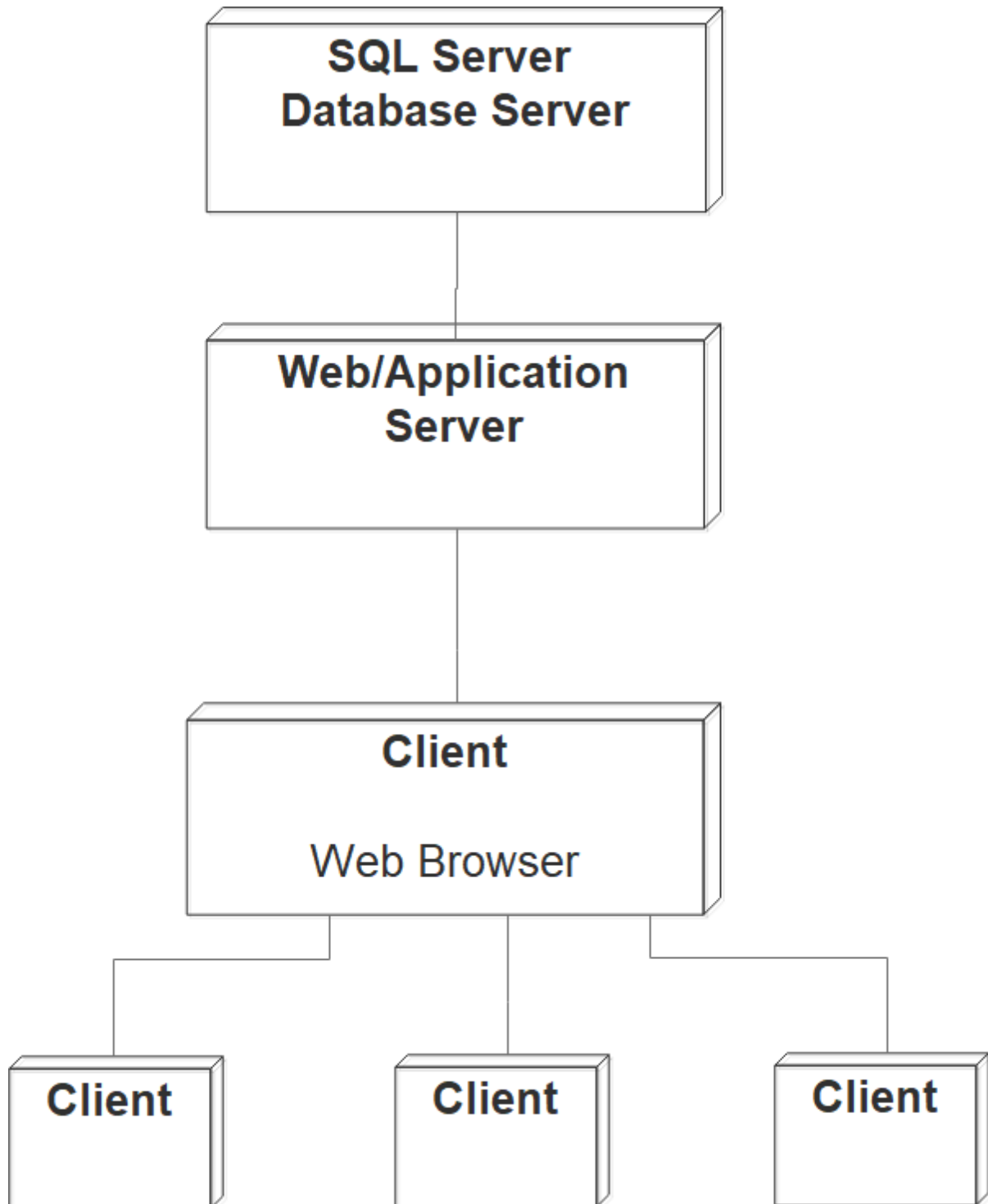
3.10. Module Implantation

1. **Administrator:**

The administrator has all the rights to access the system. He is the one who has all rights to view the applicant details, modify those details. The administrator also keeps a track of the file status of the applicants.

2. **Student:**

Student is the one who wish to visit COLLEGE ERP SYSTEM website. The student can show in his own details. The Student has rights to interaction with the COLLEGE ERP SYSTEM with giving the online examination as well as student can show the result. Student can see his absenteeism as well as they have to right to collect document. Student can check the remainder if found.

3. **Faculty:**

Faculty can show the responsible task provided by the administrator. He can view his lecture flow and other related task. Faculty can assess the exam paper and also can check the paper. He can generate his daily report to the administrator and full fill the attendance of student. Apart from that he can see his batch with its progress and evaluate of student progress. Faculty cans also sending the global note to his desired student.

3.12. Web Site Map

Sitemap

PRODUCT PAGES

Grading and Examination	**School Messenger**	**Fee Collections**	**Admissions**
Grading and Examination Features	School Messenger Features	Fee Collections Features	Admissions Features
Grading and Examinations Users	School Messenger Users	Fee Collections Users	Admissions Users
Grading and Examinations Integrations	School Messenger Integrations	Fee Collections Integrations	Admissions Resources
Grading and Examinations Resources	School Messenger Resources	Fee Collections Resources	Admissions Support
Grading and Examinations Support	School Messenger Support	Fee Collections Support	

Transport	**HR & PayRoll**	**Library**	**School Information System**
Transport Features	HR & PayRoll Features	Library Features	School Information System Features
Transport Users	HR & PayRoll Users	Library Users	School Information System Users
Transport Integrations	HR & PayRoll Integrations	Library Add-ons	School Information System Integrations
Transport Resources	HR & PayRoll Resources	Library Integrations	School Information System Resources
Transport Support	HR & PayRoll Support	Library Support	School Information System Support

Mobile Apps	**Pages**		
▪ Parent App	▪ 108 Reasons	▪ Success Stories	▪ Channel Partner
▪ Teacher App	▪ About Us	▪ Core Team	▪ Contact Sales
▪ Transport App	▪ Pricing	▪ Careers	▪ Support
▪ Accountant App	▪ Home	▪ Blog	▪ Login

3.13. User Interface Design

Admin

Faculty

Student

3.14. Data Dictionary and Database Table Design

Data Dictionary

In database management system file that defines the basic organization of a database. A data dictionary contains a list of all files in the database, the number of records in each file, and the names and types of each field. Most database management systems keep the data dictionary hidden from users to prevent them from accidentally destroying its contents.

Data dictionaries do not contain any actual data from the database, only book keeping information for managing it. Without a data dictionary, however, a database management system cannot access data from the database.

Data Dictionary

Sr.No	Field Name	Data Type	Size	Constraint	Description
1	studId	Int	15	PK	Student Identification No
2	title	Varchar	5	Not Null	Title
3	firstName	Varchar	15	Not Null	Student First name
4	middleName	Varchar	15	Not Null	Student Middle Name
5	lastName	Varchar	15	Not Null	Student last name
6	currentSession	Varchar	10	Not Null	Student Current Session
7	gender	Boolean	5	Not Null	Student Gender
8	dob	Date	5	Not Null	Student date of birth
9	age	Int	5	Not Null	Student age
10	birthPlace	Varchar	10	Not Null	Student birth place
11	religion	Varchar	10	Not Null	Student religion
12	nationality	Varchar	10	Not Null	Student nationality
13	board	Varchar	15	Not Null	Student board

14	category	Varchar	10	Not Null	Student category
15	cast	Varchar	10	Not Null	Student cast
16	motherTounge	Varchar	10	Not Null	Student mother tounge
17	class	Varchar	15	Not Null	Student class
18	userName	Varchar	15	Not Null	Student user name
19	password	Varchar	15	Not Null	Student password
20	photo	Varchar	10	Not Null	Student photo
21	bloodGroup	Varchar	10	Not Null	Student blood group
22	medium	Varchar	10	Not Null	Student medium
23	previousSchool	Varchar	15	Not Null	Student previous school
24	adharNo	Int	10	Not Null	Student adhar no
25	attended	Varchar	15	Not Null	Student attended
26	studyingClass	Int	10	Not Null	Student studying class
27	leavingDate	Date	15	Not Null	Student leaving date
28	transCertin	Varchar	10	Not Null	Transfer certificate
29	admitionDate	Date	15	Not Null	Student Admition date
30	cast	Varchar	10	Not Null	Student Cast
31	markSheet	Varchar	10	Not Null	Student arksheet
32	admitionType	Varchar	10	Not Null	Student Admition type
33	handicap	Varchar	10	Not Null	Student Handicap
34	eduGap	Varchar	15	Not Null	Student Education gap
35	resiAddress	Varchar	15	Not Null	Student Residence address
36	city	Varchar	15	Not Null	Student city

37	area	Varchar	15	Not Null	Student area
38	state	Varchar	10	Not Null	Student state
39	pincode	Int	10	Not Null	Student pin code
40	mob	Int	10	Not Null	Student mob
41	landline	Int	11	Not Null	Student landline
42	fathersMob	Int	10	Not Null	Student fathers mob
43	mothersMob		10	Not Null	Student mothers mob
44	email	Varchar	15	Not Null	Student email
45	resiAddress	Varchar	15	Not Null	Student Residence Address
46	city1	Varchar	15	Not Null	Student city
47	area1	Varchar	15	Not Null	Student area
48	state1	Varchar	15	Not Null	Student state
49	pincode1	Int	15	Not Null	Student pin code
50	mob1	Int	10	Not Null	Student Mob no
51	landline1	Int	11	Not Null	Student Landline
52	fathersMob1	Int	10	Not Null	Student fathers mob
53	mothersMob1	Int	10	Not Null	Student mothers mob
54	email1	Varchar	15	Not Null	Student email
56	fatherName1	Varchar	15	Not Null	Students father name
57	eduQuali	Varchar	15	Not Null	Student educational qualification
58	age	Int	15	Not Null	Student age
59	occupation	Varchar	15	Not Null	Student occupation
60	officeAdd	Varchar	10	Not Null	Student office address

61	motherName	Varchar	12	Not Null	Student Mother name
62	eduQuali	Varchar	12	Not Null	Student Mother Education Qualification
63	age	Int	15	Not Null	Student Mother Age
64	occupation	Varchar	15	Not Null	Student Mother Occupation
65	officeAdd	Varchar	12	Not Null	Student Mother Office Address
66	guardianName	Varchar	15	Not Null	Student Guardian Name
67	officePhone	Int	15	Not Null	Student Guardian Office Phone
68	resiAdd	Varchar	10	Not Null	Student Guardians Residence Address
69	familyMembers	Varchar	15	Not Null	Student Family Members
70	age	Int	10	Not Null	Student Family Members Age
71	relation	Varchar	15	Not Null	Student Family Members Relation
72	education	Varchar	10	Not Null	Student Family Members Education
73	occupation	Varchar	15	Not Null	Student Family Members Occupation
74	schoolName	Varchar	10	Not Null	Student Family Members School name
75	enrollment	Varchar	15	Not Null	Student Enrollment no
76	fromYear	Varchar	10	Not Null	Student Previous school From Year

77	upTo	Varchar	15	Not Null	Student Previous school till Year
78	reason	Varchar	10	Not Null	Reason of Leaving
79	loginId	Int	15	Not Null	Previous school login id
80	transport	Varchar	15	Not Null	Student Transport
81	pickupPoint	Varchar	15	Not Null	Student Pick up Point
82	route	Varchar	15	Not Null	Student Transport Route
83	otherTransport	Varchar	10	Not Null	Student other Transport Route
84	transportDesc	Varchar	15	Not Null	Transport Description
85	groupId	Int	10	PK	Group Identification No
86	studId	Int	10	FK	
87	action	Varchar	15	Not Null	Edit/Update/Delete
88	class	Int	15	Not Null	Fees Class
89	financeYear	Year	15	Not Null	Financial Year
90	particular	Varchar	15	Not Null	Fees Particular
91	amount	Int	10	Not Null	Fees Amount
92	studId	Int	10	FK	
93	paidId	Int	15	PK	Paid Fees Identification No
94	class	Varchar	15	Not Null	Paid Fees Class
95	year	Year	14	Not Null	Paid Fees
96	from	Date	15	Not Null	Paid Fees From
97	to	Date	15	Not Null	Paid Fees To
98	cateId	Int	15	PK	Category Identification No

99	category	Varchar	10	Not Null	Fees Category
100	from	Date	15	Not Null	Category From
101	to	Date	15	Not Null	Category to
102	financeYear	Year	15	Not Null	Category Financial year
103	examId	Int	15	PK	exam Identification No
104	studId	Int	10	FK	
105	examName	Varchar	15	Not Null	Exam Name
106	action	Varchar	10	Not Null	View/Update/Delete
107	group	Varchar	15	Not Null	Exam Group
108	class	Int	10	Not Null	Exam Class
109	examName	Varchar	15	Not Null	Exam Name
110	acadenicYear	Year	10	Not Null	Exam Academic Year
111	subjectId	Int	15	PK	Subject wise Marks Identification No
112	studId	Int	15	FK	
113	group	Varchar	14	Not Null	Subject Group
114	class	Int	15	Not Null	Subject Class
115	examName	Varchar	15	Not Null	Subject Exam Name
116	academicYear	Year	14	Not Null	Subject Academic Year
117	subwiseId	Int	15	PK	Student wise Marks Identification No
118	studId	Int	14	FK	
119	academicYear	Year	15	Not Null	Student Academic Year
120	class	Varchar	15	Not Null	Student Class
121	regNo	Int	10	Not Null	Student Reg No

122	examName	Varchar	15	Not Null	Student Exam Name
123	rankId	Int		PK	Rank Identification No
124	class	Int		Not Null	Rank Class
125	exam	Varchar		Not Null	Rank Exam
126	academicSession	Varchar		Not Null	Rank Academic Session
127	deptId	Int	15	PK	Department Identification No
128	studId	Int	15	FK	
129	deptName	Varchar	10	Not Null	Department Name
130	action	Varchar	15	Not Null	View/Update/Delete
131	chooseDept	Varchar	10	Not Null	Department Selection
132	staffId	Int	10	PK	Staff Identification No
133	studId	Int	10	FK	
134	employmentId	Int	15	Not Null	Staff Employment Id
135	firstName	Varchar	15	Not Null	Staff First Name
136	lastName	Varchar	15	Not Null	Staff Last name
137	gender	Boolean	10	Not Null	Staff Gender
138	fatherName	Varchar	15	Not Null	Staff Father Name
139	teachingStaff	Varchar	10	Not Null	Staff Teaching or Non Teaching
140	dept	Varchar	15	Not Null	Staff Department
141	postaApliedFor	Varchar	15	Not Null	Staff Post Applied For
142	class	Int	15	Not Null	Staff Class
143	primarySubject	Varchar	15	Not Null	Staff Primary Subject

144	uploadAsignment	Varchar	10	Not Null	Staff Upload Sign
145	markAttendance	Varchar	15	Not Null	Staff Mark attendance
146	uploadExam	Varchar	10	Not Null	Staff Upload Exam
147	obtainMarks	Int	15	Not Null	Staff Obtain marks
148	userName	Varchar	15	Not Null	Staff User name
149	password	Varchar	15	Not Null	Staff Password
150	email	Varchar	15	Not Null	Staff Email
151	nationality	Varchar	10	Not Null	Staff Nationality
152	hobbies	Varchar	10	Not Null	Staff Hobbies
153	maritalStatus	Boolean	15	Not Null	Staff marital status
154	experience	Int	15	Not Null	Staff Experience
155	attachedDocs	Varchar	15	Not Null	Staff Attached Docs
156	reservCategory	Varchar	15	Not Null	Staff Reserve category
157	bloodGroup	Varchar	15	Not Null	Staff Blood Group
158	uploadPhoto	Blob	10	Not Null	Staff Upload Photo
159	dob	Date	10	Not Null	Staff Date of Birth
160	examPassed	Boolean	10	Not Null	Staff Exam Passed
161	marksObtained	Int	10	Not Null	Staff marks Obtained
162	board	Varchar	10	Not Null	Staff Board
163	year	Varchar	10	Not Null	Staff Year
164	institute	Varchar	10	Not Null	Staff Institute
165	position	Varchar	10	Not Null	Staff position
166	period	Int	15	Not Null	Staff Period
167	address	Varchar	10	Not Null	Staff address
168	city	Varchar	15	Not Null	Staff city
169	county	Varchar	10	Not Null	Staff country

170	phone	Int	10	Not Null	Staff phone
171	address1	Varchar	15	Not Null	Staff address 1
172	city1	Varchar	15	Not Null	Staff city 1
173	county1	Varchar	15	Not Null	Staff country 1
174	phone1	Int	10	Not Null	Staff phone no 1
175	familyMember	Varchar	15	Not Null	Staff family no
176	age	Int	14	Not Null	Staff age
177	relation	Varchar	15	Not Null	Staff relation
178	education	Varchar	10	Not Null	Staff education
179	occupation	Varchar	15	Not Null	Staff occupation
180	previousPackage	Varchar	10	Not Null	Staff previous salary
181	basic	Varchar	15	Not Null	Staff basic salary
182	doj	Date	10	Not Null	Staff Date of joining
183	remarks	Varchar	15	Not Null	Staff Remarks
184	transport	Varchar	15	Not Null	Staff Transport
185	pickupPoint	Varchar	15	Not Null	Staff pick up point
186	route	Varchar	15	Not Null	Staff route

3.15. Database Table Design

1. Add Student

Sr.No	Field Name	Data Type	Size	Constraint	Description
1	studId	Int	15	PK	Student Identification No
2	title	Varchar	5	Not Null	Title
3	firstName	Varchar	15	Not Null	Student First name
4	middleName	Varchar	15	Not Null	Student Middle Name
5	lastName	Varchar	15	Not Null	Student last name
6	currentSession	Varchar	10	Not Null	Student Current Session
7	gender	Boolean	5	Not Null	Student Gender
8	dob	Date	5	Not Null	Student date of birth
9	age	Int	5	Not Null	Student age
10	birthPlace	Varchar	10	Not Null	Student birth place
11	religion	Varchar	10	Not Null	Student religion
12	nationality	Varchar	10	Not Null	Student nationality
13	board	Varchar	15	Not Null	Student board
14	category	Varchar	10	Not Null	Student category
15	cast	Varchar	10	Not Null	Student cast

16	motherTounge	Varchar	10	Not Null	Student mother tounge
17	class	Varchar	15	Not Null	Student class
18	userName	Varchar	15	Not Null	Student user name
19	password	Varchar	15	Not Null	Student password
20	photo	Varchar	10	Not Null	Student photo
21	bloodGroup	Varchar	10	Not Null	Student blood group
22	medium	Varchar	10	Not Null	Student medium
23	previousSchool	Varchar	15	Not Null	Student previous school
24	adharNo	Int	10	Not Null	Student adhar no
25	attended	Varchar	15	Not Null	Student attended
26	studyingClass	Int	10	Not Null	Student studying class
27	leavingDate	Date	15	Not Null	Student leaving date
28	transCertin	Varchar	10	Not Null	Transfer certificate
29	admitionDate	Date	15	Not Null	Student Admition date
30	cast	Varchar	10	Not Null	Student Cast
31	markSheet	Varchar	10	Not Null	Student arksheet

32	admitionType	Varchar	10	Not Null	Student Admition type
33	handicap	Varchar	10	Not Null	Student Handicap
34	eduGap	Varchar	15	Not Null	Student Education gap
35	resiAddress	Varchar	15	Not Null	Student Residence address
36	city	Varchar	15	Not Null	Student city
37	area	Varchar	15	Not Null	Student area
38	state	Varchar	10	Not Null	Student state
39	pincode	Int	10	Not Null	Student pin code
40	mob	Int	10	Not Null	Student mob
41	landline	Int	11	Not Null	Student landline
42	fathersMob	Int	10	Not Null	Student fathers mob
43	mothersMob	Int	10	Not Null	Student mothers mob
44	email	Varchar	15	Not Null	Student email
45	resiAddress	Varchar	15	Not Null	Student Residence Address
46	city1	Varchar	15	Not Null	Student city
47	area1	Varchar	15	Not Null	Student area
48	state1	Varchar	15	Not Null	Student state
49	pincode1	Int	15	Not Null	Student pin code

50	mob1	Int	10	Not Null	Student Mob no
51	landline1	Int	11	Not Null	Student Landline
52	fathersMob1	Int	10	Not Null	Student fathers mob
53	mothersMob1	Int	10	Not Null	Student mothers mob
54	email1	Varchar	15	Not Null	Student email
56	fatherName1	Varchar	15	Not Null	Students father name
57	eduQuali	Varchar	15	Not Null	Student educational qualification
58	age	Int	15	Not Null	Student age
59	occupation	Varchar	15	Not Null	Student occupation
60	officeAdd	Varchar	10	Not Null	Student office address
61	motherName	Varchar	12	Not Null	Student Mother name
62	eduQuali	Varchar	12	Not Null	Student Mother Education Qualification
63	age	Int	15	Not Null	Student Mother Age
64	occupation	Varchar	15	Not Null	Student Mother Occupation
65	officeAdd	Varchar	12	Not Null	Student Mother Office Address

66	guardianName	Varchar	15	Not Null	Student Guardian Name
67	OfficePhone	Int	15	Not Null	Student Guardian Office Phone
68	resiAdd	Varchar	10	Not Null	Student Guardians Residence Address
69	familyMembers	Varchar	15	Not Null	Student Family Members
70	age	Int	10	Not Null	Student Family Members Age
71	relation	Varchar	15	Not Null	Student Family Members Relation
72	education	Varchar	10	Not Null	Student Family Members Education
73	occupation	Varchar	15	Not Null	Student Family Members Occupation
74	schoolName	Varchar	10	Not Null	Student Family Members School name
75	enrollment	Varchar	15	Not Null	Student Enrollment no
76	fromYear	Varchar	10	Not Null	Student Previous school From Year

77	upTo	Varchar	15	Not Null	Student Previous school till Year
78	reason	Varchar	10	Not Null	Reason of Leaving
79	loginId	Int	15	Not Null	Previous school login id
80	transport	Varchar	15	Not Null	Student Transport
81	pickupPoint	Varchar	15	Not Null	Student Pick up Point
82	route	Varchar	15	Not Null	Student Transport Route
83	otherTransport	Varchar	10	Not Null	Student other Transport Route
84	transportDesc	Varchar	15	Not Null	Transport Description

2. Add Fees

Sr.No	Field Name	Data Type	Size	Constraint	Description
1	groupId	Int	10	PK	Group Identification No
2	studId	Int	10	FK	
3	action	Varchar	15	Not Null	Edit/Update/Delete
4	class	Int	15	Not Null	Fees Class
5	financeYear	Year	15	Not Null	Financial Year
6	particular	Varchar	15	Not Null	Fees Particular
7	amount	Int	10	Not Null	Fees Amount

3. Paid Fees Details

Sr.No	Field Name	Data Type	Size	Constraint	Description
1	studId	Int	10	FK	
2	paidId	Int	15	PK	Paid Fees Identification No
3	class	Varchar	15	Not Null	Paid Fees Class
4	year	Year	14	Not Null	Paid Fees
5	from	Date	15	Not Null	Paid Fees From
6	to	Date	15	Not Null	Paid Fees To

4. Category Details

Sr.No	Field Name	Data Type	Size	Constraint	Description
1	cateId	Int	15	PK	Category Identification No
2	category	Varchar	10	Not Null	Fees Category
3	from	Date	15	Not Null	Category From
4	to	Date	15	Not Null	Category to
5	financeYear	Year	15	Not Null	Category Financial year

5. Add Exam

Sr.No	Field Name	Data Type	Size	Constraint	Description
1	examId	Int	15	PK	exam Identification No
2	studId	Int	10	FK	

3	examName	Varchar	15	Not Null	Exam Name
4	action	Varchar	10	Not Null	View/Update/Delete
5	group	Varchar	15	Not Null	Exam Group
6	class	Int	10	Not Null	Exam Class
7	examName	Varchar	15	Not Null	Exam Name
8	acadenicYear	Year	10	Not Null	Exam Academic Year

6. Subject wisemarks

Sr.No	Field Name	Data Type	Size	Constraint	Description
1	subjectId	Int	15	PK	Subject wise Marks Identification No
2	studId	Int	15	FK	
3	group	Varchar	14	Not Null	Subject Group
4	class	Int	15	Not Null	Subject Class
5	examName	Varchar	15	Not Null	Subject Exam Name
6	academicYear	Year	14	Not Null	Subject Academic Year

7. Student wisemarks

Sr.No	Field Name	Data Type	Size	Constraint	Description
1	subwiseId	Int	15	PK	Student wise Marks Identification No
2	studId	Int	14	FK	

3	academicYear	Year	15	Not Null	Student Academic Year
4	class	Varchar	15	Not Null	Student Class
5	regNo	Int	10	Not Null	Student Reg No
6	examName	Varchar	15	Not Null	Student Exam Name

8. Student rank

Sr.No	Field Name	Data Type	Size	Constraint	Description
1	rankId	Int		PK	Rank Identification No
2	class	Int		Not Null	Rank Class
3	exam	Varchar		Not Null	Rank Exam
4	academicSession	Varchar		Not Null	Rank Academic Session

9. Add Department

Sr.No	Field Name	Data Type	Size	Constraint	Description
1	deptId	Int	15	PK	Department Identification No
2	studId	Int	15	FK	
3	deptName	Varchar	10	Not Null	Department Name
4	action	Varchar	15	Not Null	View/Update/Delete
5	chooseDept	Varchar	10	Not Null	Department Selection

10. Add Staff

Sr.No	Field Name	Data Type	Size	Constraint	Description

1	staffId	Int	10	PK	Staff Identification No
2	studId	Int	10	FK	
3	emplymentId	Int	15	Not Null	Staff Employment Id
4	firstName	Varchar	15	Not Null	Staff First Name
5	lastName	Varchar	15	Not Null	Staff Last name
6	gender	Boolean	10	Not Null	Staff Gender
7	fatherName	Varchar	15	Not Null	Staff Father Name
8	teachingStaff	Varchar	10	Not Null	Staff Teaching or Non Teaching
9	dept	Varchar	15	Not Null	Staff Department
10	postAppliedFor	Varchar	15	Not Null	Staff Post Applied For
11	class	Int	15	Not Null	Staff Class
12	primarySubject	Varchar	15	Not Null	Staff Primary Subject
13	uploadAsignment	Varchar	10	Not Null	Staff Upload Sign
14	markAttendance	Varchar	15	Not Null	Staff Mark attendance
15	uploadExam	Varchar	10	Not Null	Staff Upload Exam
16	obtainMarks	Int	15	Not Null	Staff Obtain marks
17	userName	Varchar	15	Not Null	Staff User name

18	password	Varchar	15	Not Null	Staff Password
19	email	Varchar	15	Not Null	Staff Email
20	nationality	Varchar	10	Not Null	Staff Nationality
21	hobbies	Varchar	10	Not Null	Staff Hobbies
22	maritalStatus	Boolean	15	Not Null	Staff marital status
23	experience	Int	15	Not Null	Staff Experience
24	attachedDocs	Varchar	15	Not Null	Staff Attached Docs
25	reservCategory	Varchar	15	Not Null	Staff Reserve category
26	bloodGroup	Varchar	15	Not Null	Staff Blood Group
27	uploadPhoto	Blob	10	Not Null	Staff Upload Photo
28	dob	Date	10	Not Null	Staff Date of Birth
29	examPassed	Boolean	10	Not Null	Staff Exam Passed
30	marksObtained	Int	10	Not Null	Staff marks Obtained
31	board	Varchar	10	Not Null	Staff Board
32	year	Varchar	10	Not Null	Staff Year
33	institute	Varchar	10	Not Null	Staff Institute
34	position	Varchar	10	Not Null	Staff position
35	period	Int	15	Not Null	Staff Period
36	address	Varchar	10	Not Null	Staff address
37	city	Varchar	15	Not Null	Staff city

38	county	Varchar	10	Not Null	Staff country
39	phone	Int	10	Not Null	Staff phone
40	address1	Varchar	15	Not Null	Staff address 1
41	city1	Varchar	15	Not Null	Staff city 1
42	county1	Varchar	15	Not Null	Staff country 1
43	phone1	Int	10	Not Null	Staff phone no 1
44	familyMember	Varchar	15	Not Null	Staff family no
45	age	Int	14	Not Null	Staff age
46	relation	Varchar	15	Not Null	Staff relation
47	education	Varchar	10	Not Null	Staff education
48	occupation	Varchar	15	Not Null	Staff occupation
49	previousPackage	Varchar	10	Not Null	Staff previous salary
50	basic	Varchar	15	Not Null	Staff basic salary
51	doj	Date	10	Not Null	Staff Date of joining
52	remarks	Varchar	15	Not Null	Staff Remarks
53	transport	Varchar	15	Not Null	Staff Transport
54	pickupPoint	Varchar	15	Not Null	Staff pick up point
55	route	Varchar	15	Not Null	Staff route

3.16. TESTING PPROCEDURE AND IMPLEMENTATION

Software Testing has a dual function; it is used to identify the defects in program and it is used to help judge whether or not program is usable in practice. Thus software testing is used for validation and verification, which ensure that software conforms to its specification and meets need of the software customer.

Developer resorted Alpha testing, which usually comes in after the basic design of the program has been completed. The project scientist will look over the program and give suggestions and ideas to improve or correct the design. They also report and give ideas to get rid of around any major problems. There is bound to be a number of bugs after a program have been created.
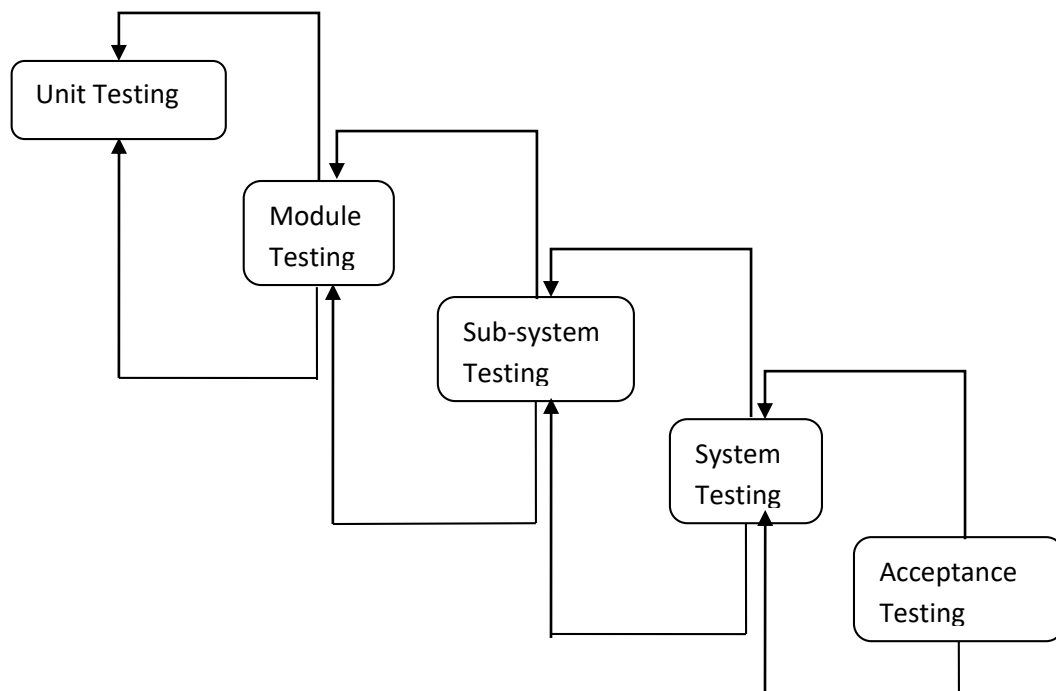
Figure 1 Testing Step

Branch coverage

It is strategy in which test cases are designed to make each branch condition assume true & false values.

Conditional coverage

In this testing test cases are designed to make each component of composite conditional expression both true & false.

The Testing Process

We test the software process activities such as Design, Implementation, and Requirement Engineering. Because, design errors are very costly to repair once system has been started to operate, it is quite obvious to repair them at early stage of the system. So analysis is the most important process of any project.

Requirement Traceability

As most interested portion is whether the system is meeting its requirements or not, for that testing should be planned so that all requirements are individually tested. We checked the output of certain combination of inputs so that we can know whether it gives desirable results or not. Strictly sticking to your requirements specifications, give you the path to get desirable results from the system.

4. Testing Schedule

We have tested each procedure back-to-back so that errors and omissions can be found as early as possible. Once the system has been developed fully we tested it on other machines, which differs in configuration.

TESTING STRATEGY

There are types of testing that we implement. They are as follows:

- While deciding on the focus of testing activities, study project priorities. For example, for an on-line system, pay more attention to response time. Spend more time on the features used frequently.
- Decide on the effort required for testing based on the usage of the system. If the system is to be used by a large number of users, evaluate the impact on users due to a system failure before deciding on the effort.

- A necessary part of the test case is a definition of the expected result.

- Write test cases for invalid and unexpected as well as valid and expected input conditions.

- Thoroughly inspect the results of each test.

We have performed both Unit Testing and System Testing on WIMS to detect and fix errors. A brief description of both is given below.

5. White Box Testing

White-box testing (also known as **clear box testing**, **glass box testing**, **transparent box testing**, and **structural testing**) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

White-box test design techniques include:

- Control flow testing
- Data flow testing
- Branch testing
- Path testing
- Statement coverage
- Decision coverage

Overview

White-box testing is a method of testing the application at the level of the source code. The test cases are derived through the use of the design techniques mentioned above: control

flow testing, data flow testing, branch testing, path testing, statement coverage and decision coverage as well as modified condition/decision coverage.

White-box testing is the use of these techniques as guidelines to create an error free environment by examining any fragile code. These White-box testing techniques are the building blocks of white-box testing, whose essence is the careful testing of the application at the source code level to prevent any hidden errors later on. These different techniques exercise every visible path of the source code to minimize errors and create an error-free environment. The whole point of white-box testing is the ability to know which line of the code is being executed and being able to identify what the correct output should be.

Levels

1. Unit testing. White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.
2. Integration testing. White-box testing at this level is written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.
3. Regression testing. White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

Basic Procedure

White-box testing's basic procedures involve the understanding of the source code that you are testing at a deep level to be able to test them. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once the source code is understood then the source code can be analyzed

for test cases to be created. These are the three basic steps that white-box testing takes in order to create test cases:

Input, involves different types of requirements, functional specifications, detailed designing of documents, proper source code, security specifications. This is the preparation stage of white-box testing to layout all of the basic information.

Processing Unit involves performing risk analysis to guide whole testing process, proper test plan, execute test cases and communicate results. This is the phase of building test cases to make sure they thoroughly test the application the given results are recorded accordingly.

Output, prepare final report that encompasses all of the above preparations and results.

Advantages

White-box testing is one of the two biggest testing methodologies used today. It primarily has three advantages:

1. A side effect of having the knowledge of the source code is beneficial to thorough testing.
2. Optimization of code by revealing hidden errors and being able to remove these possible defects.
3. Gives the programmer introspection because developers carefully describe any new implementation.

Disadvantages

Although White-box testing has great advantages, it is not perfect and contains some disadvantages. It has two disadvantages:

1. White-box testing brings complexity to testing because to be able to test every important aspect of the program, you must have great knowledge of the program. White-box testing requires a programmer with a high-level of knowledge due to the complexity of the level of testing that needs to be done.

2. On some occasions, it is not realistic to be able to test every single existing condition of the application and some conditions will be untested.

6.   Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

Test Procedures

Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of *what* the software is supposed to do but is not aware of *how* it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of *how* the software produces the output in the first place.

Test Cases

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily *functional* in nature, *non-functional* tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output without any knowledge of the test object's internal structure.

Test Design Techniques

Typical black-box test design techniques include:

- Decision table testing
- All-pairs testing
- State transition Analysis
- Equivalence partitioning
- Boundary value analysis
- Cause–effect graph
- Error guessing

Advantages

- Efficient when used on large systems.
- Since the tester and developer are independent of each other, testing is balanced and unprejudiced.
- Tester can be non-technical.
- There is no need for the tester to have detailed functional knowledge of system.
- Tests will be done from an end user's point of view, because the end user should accept the system. (This testing technique is sometimes also called Acceptance testing.)
- Testing helps to identify vagueness and contradictions in functional specifications.
- Test cases can be designed as soon as the functional specifications are complete.

Disadvantages

- Test cases are challenging to design without having clear functional specifications.
- It is difficult to identify tricky inputs if the test cases are not developed based on specifications.
- It is difficult to identify all possible inputs in limited testing time. As a result, writing test cases may be slow and difficult.
- There are chances of having unidentified paths during the testing process.
- There is a high probability of repeating tests already performed by the programmer.

1] Test case For Admin Login Page:

Project Name: COLLEGE ERP SYSTEM

Prepared Date:-21-12-2020.   Prepared By:-Aneri

Module Name: Login.   Reviewed Date:-21-12-2020

Project Code: - COLLEGEERPSYSTEM.   Reviewed By:-

Total no of test Cases:-04

Total no of test Cases Passed:-04

Total no of test Cases failed:-00

Total no of test Cases executed:-04

Total no of test Cases pending:-00

Test Case ID	Test Case Procedure	Input Data	Expected Output	Actual Output	Test Status
SSTUDE NT MANAG EMENT SYSTEM -LG-01	Checking the functionality of Admin LOGIN Button	1.Enter valid Usernames in textbox 2. Enter valid Password in textbox 3. Click on Admin LOGIN Button	Admin Welcome page should be displayed	Admin Welcome page displayed	Pass
SSTUDE NT MANAG EMENT SYSTEM -LG-02	Checking the functionality of Admin LOGIN Button	1.Enter invalid User Name in text box 2. Enter valid Password in password textbox	Admin Welcome page should not be displayed	Admin Welcome page is not displayed	Pass

		3. Click on Admin LOGIN Button			
SCOLLEGE ERP SYSTEM -LG-03	Checking the functionality of Admin LOGIN Button	1.Enter valid User name in User name textbox 2. Enter invalid Password in password textbox 3. Click on Admin LOGIN Button	Admin Welcome page should not be displayed	Admin Welcome page is not displayed	Pass
SCOLLEGE ERP SYSTEM -LG-04	Checking the functionality of Admin LOGIN Button	1.Enter invalid User name in User name textbox 2. Enter invalid Password in password textbox 3. Click on Admin LOGIN Button	Admin Welcome page should not be displayed	Admin Welcome page is not displayed	Pass

2] Test case Register Student Page:

Project Name: COLLEGE ERP SYSTEM

Prepared Date:- 21-12-2020.   Prepared By:- Aneri

Module Name: Manage Student.   Reviewed Date:- 21-12-2020

Project Code: - COLLEGEERPSYSTEM.   Reviewed By:-

Total no of test Cases:-04

Total no of test Cases Passed:-04

Total no of test Cases failed:-00

Total no of test Cases executed:-04

Total no of test Cases pending:-00

Test Case ID	Test Case Procedure	Input Data	Expected Output	Actual Output	Test Status
SCOLLEGE ERP SYSTEM-MS-01	Checking the functionality of Add Student Button	1.Enter valid Student Name in textbox 2. Enter Personal Detail in Text box 3. Enter valid Education Detail in text box. 4. Click on Add Student Button	Student Registered Successfully should be displayed	Student Registered Successfully displayed	Pass
SCOLLEGE ERP SYSTEM-MS-02	Checking the functionality of Add Student Button	1.Enter valid Student Name in textbox 2. Enter Invalid Personal Detail in Text box 3. Enter valid Education Detail in text box. 4. Click on Add Student Button	Student Registered Successfully should not be displayed	Student Registered Successfully not displayed	Pass

SCOLLEG E ERP SYSTEM-MS-03	Checking the functionality of Add Student Button	1.Enter valid Student Name in textbox 2. Enter Personal Detail in Text box 3. Enter Invalid Education Detail in text box. 4. Click on Add Student Button	Student Registered Successfully should not be displayed	Student Registered Successfully not displayed	Pass
SCOLLEG E ERP SYSTEM-MS-04	Checking the functionality of Add Student Button	1.Enter Invalid Student Name in textbox 2. Enter Invalid Personal Detail in Text box 3. Enter Invalid Education Detail in text box. 4. Click on Add Student Button	Student Registered Successfully should not be displayed	Student Registered Successfully not displayed	Pass

Test Design Techniques

Typical black-box test design techniques include:

- Decision table testing
- All-pairs testing
- State transition Analysis
- Equivalence partitioning
- Boundary value analysis

- Cause–effect graph
- Error guessing

Advantages

- Efficient when used on large systems.
- Since the tester and developer are independent of each other, testing is balanced and unprejudiced.
- Tester can be non-technical.
- There is no need for the tester to have detailed functional knowledge of system.
- Tests will be done from an end user's point of view, because the end user should accept the system. (This testing technique is sometimes also called Acceptance testing.)
- Testing helps to identify vagueness and contradictions in functional specifications.
- Test cases can be designed as soon as the functional specifications are complete.

Disadvantages

- Test cases are challenging to design without having clear functional specifications.
- It is difficult to identify tricky inputs if the test cases are not developed based on specifications.
- It is difficult to identify all possible inputs in limited testing time. As a result, writing test cases may be slow and difficult.
- There are chances of having unidentified paths during the testing process.
- There is a high probability of repeating tests already performed by the programmer.

7. Unit Testing

In computer programming, **unit testing** is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process.

Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

Benefits

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

Unit tests find problems early in the development cycle.

In test-driven development (TDD), which is frequently used in both Extreme Programming and Scrum, unit tests are created before the code itself is written. When the tests pass, that code is

considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

Facilitates change

Unit testing allows the programmer to refactor code at a later date, and make sure the module still works correctly. The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified.

Readily available unit tests make it easy for the programmer to check whether a piece of code is still working properly.

In continuous unit testing environments, through the inherent practice of sustained maintenance, unit tests will continue to accurately reflect the intended use of the executable and code in the face of any change. Depending upon established development practices and unit test coverage, up-to-the-second accuracy can be maintained.

Si**mplifies in**tegration

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

An elaborate hierarchy of unit tests does not equal integration testing. Integration with peripheral units should be included in integration tests, but not in unit tests. Integration testing typically still relies heavily on humans testing manually; high-level or global-scope testing can be difficult to automate, such that manual testing often appears faster and cheaper.

Documentation

Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit's interface.

Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviours that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development.

By contrast, ordinary narrative documentation is more susceptible to drifting from the implementation of the program and will thus become outdated (e.g., design changes, feature creep, relaxed practices in keeping documents up-to-date).

Design

When software is developed using a test-driven approach, the combination of writing the unit test to specify the interface plus the refactoring activities performed after the test is passing, may take the place of formal design. Each unit test can be seen as a design element specifying classes, methods, and observable behaviour.

Limitations

Testing will not catch every error in the program, since it cannot evaluate every execution path in any but the most trivial programs. The same is true for unit testing. Additionally, unit testing by definition only tests the functionality of the units themselves. Therefore, it will not catch integration errors or broader system-level errors (such as functions performed across multiple units, or non-functional test areas such as performance). Unit testing should be done in conjunction with other software testing activities, as they can only show the presence or absence of particular errors; they cannot prove a complete absence of errors. In order to guarantee correct behaviour for every execution path and every possible input, and ensure the absence of errors, other techniques are required, namely the application of formal methods to proving that a software component has no unexpected behaviour.

8.   Integration Testing

Integration testing (sometimes called **integration and testing**, abbreviated **I&T**) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

Purpose

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.

Some different types of integration testing are big bang, top-down, and bottom-up. Other Integration Patterns are: Collaboration Integration, Backbone Integration, Layer Integration, Client/Server Integration, Distributed Services Integration and High-frequency Integration.

Big Bang

In this approach, all or most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. The Big Bang method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.

A type of Big Bang Integration testing is called **Usage Model testing**. Usage Model Testing can be used in both software and hardware integration testing. The basis behind this type of integration testing is to run user-like workloads in integrated user-like environments. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. Usage Model testing takes an optimistic approach to testing, because it expects to have few problems with the individual components. The strategy relies heavily on the component developers to do the isolated unit testing for their product. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the components in the environment. For integration testing, Usage Model testing can be more efficient and provides better test coverage than traditional focused functional integration testing. To be more efficient and accurate, care must be used in defining the user-like workloads for creating realistic scenarios in exercising the environment. This gives confidence that the integrated environment will work as expected for the target customers.

Top-down and Bottom-up

Bottom up Testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.

Top down Testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.

Sandwich Testing is an approach to combine top down testing with bottom up testing.

The main advantage of the Bottom-Up approach is that bugs are more easily found. With Top-Down, it is easier to find a missing branch link.

Validation Testing

Validations are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it fulfils its intended purpose. The words "verification" and "validation" are sometimes preceded with "Independent" (or IV&V), indicating that the verification and validation is to be performed by a disinterested third party.

Overview

Validation is intended to ensure a product, service, or system (or portion thereof, or set thereof) result in a product, service, or system (or portion thereof, or set thereof) that meets the operational needs of the user. For a new development flow or verification flow, validation procedures may involve modelling either flow and using simulations to predict faults or gaps that might lead to invalid or incomplete verification or development of a product, service, or system (or portion thereof, or set thereof). A set of validation requirements (as defined by the user), specifications, and regulations may then be used as a basis for qualifying a development flow or verification flow for a product, service, or system (or portion thereof, or set thereof). Additional validation procedures also include those that are designed specifically to ensure that modifications made to an existing qualified development flow or verification flow will have the effect of producing a product, service, or system (or portion thereof, or set thereof) that meets the initial design requirements, specifications, and regulations; these validations help to keep the flow qualified. It is a process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders. This is often an external process.

Aspects of Validation

The most tested attributes in validation tasks may include, but are not limited to

- Selectivity/specificity
- Accuracy and precision
- Repeatability
- Reproducibility
- Limit of detection – especially for trace elements
- Limit of quantification
- Curve fitting and its range
- System suitability

To solve this kind of difficulties, some regulatory bodies or compendia methods usually provide the advices on what the circumstances or conditions that the performing of a specified system suitability test should be bearded and compulsory.

CHAPTER 4

USER MANUAL

4.1 User Manual

1. for Administrator

The User Manual describes the use of the system to Administrator. It describes the use of the system on social media. The user manual should be available

The administrator has all the rights to access the system. He is the one who has all rights to view the applicant details, modify those details. The administrator also keeps a track of the file status of the applicants.

As help. .

- Admin first login to the system
- Manage System
- Manage Applicant Details.
- View product.
- Logout

2. User Manual for Project Manager

Project managers are responsible for day-to-day project management tasks such as creating, maintaining, and updating schedules, and coordinating with other project managers, resource managers, and team members. The Project Access home page is the primary entry point for users who work with data saved to Mysql database. When you log on to system Access, pending items that might require action (such as task updates that must be approved) and items that have changed since the last time that you logged on are displayed. From the home page, you

can access Project Web Access features, including Tasks, Project Center, Resource Center, Updates, Status Reports, Documents, Issues, and Risks pages. The Project Center provides a convenient way for project managers, team members, and other project stakeholders to view detailed information about individual projects and project proposals (proposal: A suggested project plan, initiated in Project Web Access. Proposals provide better high-level analysis, and enable you to apply your business processes to track potential projects.), and to view summary information about projects across the organization.

3. User Manual for Team Leader

Team Leader are responsible for accessing day-to-day information of the system. Tasks such as monitoring, maintaining, and updating , and coordinating with other project managers and team member. Following is the user manual of the team leader. It is important task of monitoring with its feeds from site to database. Only administrator is having access to this menu. The menu will provide functionalities which are as follows:

- Managing networks details like adding new account with its standard. Modifying details of added account.
- Adding details like team member and foul words.
- Modifying details of SMP profiles and Setting.

4. User Manual for Developer

The Developer is responsible for accessing all the functionality of the system.

- Keeps monitoring that all modules are working fine.
- Server response time for the system.
- Select the task from submenus like network inbox, smp list, rss updates etc

4.2. **Menu Explanation**

1. **Admin**

1. Admin first login to the system
2. Manage System
3. Manage Applicant Details.
4. View product.
5. Logout

2. **Student**

2.1 Home

2.2 View Profile

2.3 Give Online Examination

2.4 View Result

2.5 View absenteeism

2.6 Check Documents

3. **Faculty**

3.1 Home

3.2 View Task

3.3 View Lecture FLOW

3.4 Access Exam Paper

3.5 Check paper

3.6 Manage daily report

3.7 Manage attendance

3.8 Manage student evaluation test

3.9 Manage to sending global note

4.3. **Program Specification**

Module: add Exam Details

Program Name	Constraint	Description
List of Exam Details	Admin should have unique login id and password and select/upload Exam Details.	Displayed added Exam Details

Module: View Exam Details

Program Name	Constraint	Description
View Exam Details	User should have login id password	Displays the details of the Exam Details

Module: add task

Program Name	Constraint	Description
List of task for faculty	Admin should have unique login id and password and select/upload task for faculty.	Displayed task for faculty

LIMITATIONS

The limitation of the application is more improve in online examination. Limited questions had been stored and need more updation and maintenance of the application.

 Storage capacity too small so that it cannot be stored large amount of data so take back up is necessary for the future requirement. So limited amount of data can be stored.

A Print of the result cannot print out in this application.

PROPOSED ENHANCEMENT

- Adding module for Multilanguage support.
- Adding Course comparison for Future Mind.
- Adding Annual Function Video view in video format

FUTURE ENHANCEMENTS

Make online exam more effective, efficient and more dynamic so that it helps to get a good support from the student.

It has been marking while this application open in other browser due to designing support. It's more support to Mozilla as compare to other browser.

Online result printing and more user interaction functionality and features need to be developed.

Needs to improve more security for the purpose of the safety.

Conclusion

No project can be termed as 'perfect' in real sense and there always remains scope for future improvement and so that helps to develop a new version of the software. We are always eager to know some new points and validation related to projects which give us more knowledge and help us to create new version.

The *"COLLEGE ERP SYSTEM"* system has been developed by me and also done enhancement in application through applying our knowledge gained in class room, referring to certain books, browsing some sites and through the help of external and internal faculties and using our knowledge related to subject it.

I am very thankful to the project guide and organization staffs that extended all their support and helped us complete this project successfully.

Reference

WEBSITES

- http://books.google.com
- http://sourceforge.net
- http://www.gnu.org
- http://www.thefreecountry.com

BOOKS

- **.Net Reference: Beginner**
- **Advanced .Net**

Annexures

Reports Overview:

- ❖ Student Info Report – Over here all the data related to the student personal information is been visible

- ❖ Student Progress Report – Over here student progress is been visible

- ❖ Faculty Details – Faculty personal information is been visible in this report

Sample Code

<mark>Admin View Staff/Faculty:</mark>

```
<%@ Page Title="" Language="C#" MasterPageFile="~/admin/admin.master"
AutoEventWireup="true"

  CodeFile="admin_feculty.aspx.cs" Inherits="admin_admin_feculty" %>


<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1 runat="erver">

  <center>

    <table width="100%" cellpadding="0" cellspacing="7" border="0">

      <tr>

        <td colspan="3" style="background-color: Blue">

          <asp:Label ID="Label1" runat="server" Text="Feculy Information" ForeColor="White"

            Font-Size="15pt"></asp:Label>

        </td>

      </tr>

      <tr>

        <td colspan="3">

          <hr style="background-color: # Blue; height: 0.em">

        </td>

      </tr>

      <tr>

        <td style="width: 35%">


              
  
```

          Enter ID

      </td>

      <td style="width: 5%>

          :

      </td>

      <td align="left" style="width: 60%">

          <asp:TextBox ID="TextBox7" runat="server" Width="210px"></asp:TextBox>

      </td>

    </tr>

    <tr>

      <td style="width: 35%">

          Feculty Name

      </td>

      <td style="width: 5%">

          :

      </td>

      <td align="left" style="width: 60%">

          <asp:TextBox ID"TextBox1" runat="server" Width="210px"></asp:TextBox>

      </td>

    </tr>

    <tr>

      <td style="width: 35%">

  Search

          Name

```
        </td>

        <td style="width: 5%">

            :

        </td>

        <td align="left" style="width: 60%">

            <asp:TextBox ID="TextBox2" runat="server" Width="210px"></asp:TextBox>

        </td>

    </tr>

    <tr>

        <td style=width: 35%">

              
  

            Qualification

        </td>

        <td style="width: 5%">

            :

        </td>

        <td align="left" style"width: 60%">

            <asp:TextBox ID="TextBox3" runat="server" Width="210px"></asp:TextBox>

        </td>

    </tr>

    <tr>

        <td style="width: 35%">

              
  

            Designation

        </td>
```

```
            <td style="width: 5%">

                :

            </td>

            <td align="left" style="width: 60%">

                <asp:TextBox ID="TextBox4" runat="server" Width="210px"></asp:TextBox>

            </td>

        </tr>

        <tr>

            <td style="width: 35%">
```



```
                Experiance

            </td>

            <td style="width: 5%">

                :

            </td>

            <td align="eft" style="width: 60%">

                <asp:TextBox ID="TextBox6" runat="server" Width="210px"></asp:TextBox>

            </td>

        </tr>

        <tr>

            <td style="width: 35%">
```



```
                Area OF Interest

            </td>

            <td style="width: 5%">
```

```
                :

          </td>

          <td align="left" style="width: 60%">

             <asp:TextBox ID="TextBox5" runa="server" Width="210px"></asp:TextBox>

          </td>

       </tr>

       <tr>

          <td style="width: 35%">
```



```
             Select Image

          </td>

          <td style="width: 5%">

             :

          </td>

          <td align="left" style="width: 60%">

             <asp:FileUpload ID="FileUpload1" runat="server" />

          </td>

       </tr>

       <tr>

          <td colspan="3">

             <hr style="background-color: #363258; height: 0.2em">

          </td>

       </tr>

       <tr>

          <td colspan="3">
```

```
            <asp:Button ID="Button1" runa="server" Text="Store" Width="81px"
OnClick="Button1_Click"
/>           <asp:Button

                ID="Button2" runat="server" Text="Update" Width="88px" OnClick="Button2_Click"
/>           <asp:Button

                 ID="Button3" runat="server" Text="Delete" Width="102px" OnClick="Button3_Click"
/>           <asp:Button

                  ID="Button4" runat="server" Text="Clear" Width="102px"
OnClick="Button4_Click" />

        </td>

     </tr>

     <tr>

       <td colspan="3">

         <hr style="background-color: #363258; height: 0.2em">

       </td>

     </tr>

    </table>

  </center>

</asp:Content>
```

Admin add Departments:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/admin/admin.master"
AutoEventWireup="true" CodeFile="admin_department.aspx.cs"
Inherits="admin_admin_department" %>


<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat=Server">

<center>
```

```
<table width="100%" cellpadding="0" cellspacing="7" border="0">

  <tr>

    <td colspan="3" style="background-color: #368">

      <asp:Label ID="Label1" runat="server" Text="Cources Information"></asp:Label>

    </td>

  </tr>

  <tr>

    <td colspan="3">

      <hr style="background-color: #363258; height: 0.2em">

    </td>

  </tr>

  <tr>

    <td style="width: 35%">


    &nbs;           

        Enter ID

    </td>

    <td style="width: 5%">

      :

    </td>

    <td align="left" style="width: 60%">

      <asp:TextBox ID="extBox4" runat="server" Width="210px"></asp:TextBox>

    </td>

  </tr>

  <tr>

    <td style="width: 35%">
```


    Department

  </td>

  <td style="width: 5%>

   :

  </td>

  <td align="left" style="width: 60%">

   <asp:TextBox ID="Texx1" runat="server" Width="210px"></asp:TextBox>

  </td>

 </tr>

 <tr>

  <td colspan="3">

   <hr style="background-color: #3632; height: 0.2em">

  </td>

 </tr>

 <tr>

  <td colspan="3">

   <asp:Button ID="Button" runat="server" Text="Stre" Width="81px"

   onclick="Button1_Click"
/>   &nbs;
;   </td>

 </tr>

 <tr>

  <td colspan="3">

   <hr style="background-color: #3632; height: 0.2em">

  </td>

```
        </tr>

        <tr>

          <td cospan="3" align="center">

            <asp:GridView ID="GridView1" runat"server" BackColor="Whte"

              BorderColor="999" BorderStyle="Solid" BorderWidth="1px" CellPadding="3"

              ForeColor="Blak" GridLines="Vertical" AllowPaging="True"

              AutoGenerateColumns="False" DataKeyNames="deptid"
DataSourceID="SqlDataSouce1">

              <AlternatingRowStyle BackColor="#CCCCCC" />

              <Columns>

                <asp:CommandField ShowDeleteButton="True" ShowEditButton="True"

                  ShowSelectButton="True" />

                <asp:BoundField DataField="deptid" HeaderText="deptid" InsertVisible="False"

                  ReadOnly="True" SotExpression="deptid" />

                <asp:BoundField DataField="deptname" HeaderText="deptname"

                  SortExpression="deptname" />

              </Columns>

              <FooterStyle BackColor="#CCCCCC" />

              <HeaderStyle BackColor="Black" Font-Bold="True" ForeColor="White" />

              <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />

              <SelectedRowStyle BackColor="#000099" Font-Bold="True" ForeColor="White" />

              <SortedAscendingCelStyle BackColor="#F1F1F1" />

              <SortedAscendingHeaderStyle BackColor="#808080" />

              <SortedDescendingCellStyle BackColor="#CAC9C9" />

              <SortedDescendingHeaderStyle BackColor="#383838" />

            </asp:GridView>

            <asp:SqlDataSource ID="SqlDataSource1" runat="server"

              ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
```

```
DeleteCommand="DELETE FROM [department] WHERE [deptid] = @deptid"

InsertCommnd="INSERT INTO [department] ([deptname]) VALUES (@deptname)"

SelectCommand="SELECT * FROM [department]"

UpdateCommand="UPDATE [department] SET [deptname] = @deptname WHERE
[deptid] = @deptid">

    <DeleteParameters>

       <asp:Parameter Name="deptid" Type="Int32" />

    </DeleteParameters>

    <InsertParameters>

       <asp:Parameter Name="deptname" Type="String" />

    </InsertParaeters>

    <UpdateParameters>

       <asp:Parameter Name="deptname" Type="String" />

       <asp:Parameter Name="deptid" Type="Int32" />

    </UpdateParameters>

  </asp:SqlDataSource>

 </td>

 </tr>

  </table>

  </center>

</asp:Content>
```

Admin Get Student Information:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/admin/admin.master"
AutoEventWireup="true"
```

```
    CodeFile="stu_info.asp.cs" Inherits="admin_stu_info" %>


<asp:Content ID="Content1" ContentPlaceolderID="head" runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="Server">

  <center>

    <table width="100%" cellpaddng="0" cellspacing="7" border="0">

      <tr>

        <td colspan="3" style="background-color: #363258">

          <asp:Label ID="Label1" runat="server" Text="Student Profile Display" ForeColor="White"

            Font-Size="16pt"></asp:Label>

        </td>

      </tr>

      <tr>

        <td colspan="3">

          <hr style="backgrond-color: #363258; height: 0.2em">

        </td>

      </tr>

      <tr>

        <td style="width: 35%">


    &nbs;           

          Enter ID

        </td>

        <td style="width: 5%">

          :

        </td>
```

```
    <td align="left" style="width: 60%">

        <asp:TextBox ID="TextBox3" runat="server" Width="210px"></asp:TextBox>

    </td>

</tr>

 <tr>

    <td colspan="3">

        <hr style="backgrond-color: #363258; height: 0.2em">

    </td>

</tr>

<tr>

    <td style="width: 35%">
```



```
        Student Name

    </td>

    <td style="widt: 5%">

        :

    </td>

    <td align="left" style="width: 60%>

        <asp:TextBox ID="TextBox1" runat="server" Width="210px"></asp:TextBox>

    </td>

</tr>

<tr>

    <td style="width: 35%">
```

     &bsp;

```
        Enrollment No.
```

```
            </td>

            <td style="width: 5%">

                :

            </td>

            <td align="left" style="width: 60%">

                <asp:TextBox ID="TextBox2" runat="server" Width="210px"></asp:TextBox>

            </td>

        </tr>

        <tr>

            <td style="widt: 35%">
```



```
                Branch

            </td>

            <td style="width: 5%">

                :

            </td>

            <td align="left" style="width: 60%">

                <asp:TextBox ID="TextBox4" runat="server" Width="210px"></asp:TextBox>

            </td>

        </tr>

        <tr>

            <td style="width: 35%">
```



```
                Admission Year

            </td>
```

```
      <td style="widh: 5%">

        :

      </td>

      <td align="left" style="width: 60%">

        <asp:TextBox ID="TextBox5" runat="server" Width="210px"></asp:TextBox>

      </td>

    </tr>

    <tr>

      <td colspan="3">

        <hr style="backgroun-color: #363258; height: 0.2em">

      </td>

    </tr>

    <tr>

      <td colspan="3">

        <asp:Button ID="Button1" runat="server" Text="Store" Width="81px"
OnClick="Button1_Click"
/>    &nsp;      <asp:Button

          ID="Button2" runat="server" Text="Update" Width="88px" OnClick="Button2_Click"
/>           <asp:Button

          ID="Button3" runat="srver" Text="Clear" Width="81px" OnClick="Button3_Click" />

      </td>

    </tr>

    <tr>

      <td colspan="3">

        <hr style="background-color: #363258; height: 0.2em">

      </td>

    </tr>

    <tr>

      <td colspan="3" align="center">
```

```
            <asp:GridView ID="GridView1" runat="server" BackColor="White"

                BorderColor="#999999" BorderStyle="Solid" BorderWidth="1px" CellPadding="3"
CellSpacing="4"

                ForeColor="Black" GridLines="Vertical">

                <AlternatingRowStye BackColor="#CCCCCC" />

                <FooterStyle BackColor="#CCCCCC" />

                <HeaderStyle BackColor="Black" Font-Bold="True" ForeColor="White" />

                <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />

                <SelectedRowStyle BackColor="#000099" Font-Bold="True" ForeColor="White" />

                <SortedAscendingCellStyle BackColor="#F1F1F1" />

                <SortedAscendingHeaderStyle BackColor="#808080" />

                <SortedDescendingCellStyle BackColor="#CAC9C9" />

                <SortedDescendinHeaderStyle BackColor="#383838" />

            </asp:GridView>

        </td>

      </tr>

    </table>

  </center>

</asp:Content>
```

Manage College Event:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/admin/admin.master"
AutoEventWireup="true" CodeFile="admin_newsevents.aspx.cs"
Inherits="admin_admn_newsevents" %>


<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>
```

```
<asp:Content ID="Content2" ConentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<center>

    <table width="100%" cellpadding="0" cellspacing="7" border="0">

      <tr>

        <td colspan="3" style="background-color: #363258">

          <p style="clor: White">

            <font size="14">File Upload Section</font></p>

        </td>

      </tr>

      <tr>

        <td colspan="3">

          <hr style="bakground-color: #363258; height: 2em">

        </td>

      </tr>

       <tr>

        <td style="width: 35%">


   nbsp;            

          Enter ID

        </td>

        <td style="wdth: 5%">

           :

        </td>

        <td align="left" style="width: 60%">

          <asp:TextBox ID="TextBox4" runat="server" Width="210px"></asp:TextBox>

        </td>

      </tr>
```

```
<tr>            <td style="width: 35%">
```



```
        Enter File Title

      </td>

      <td style="width: 5%">

        :

      </td>

      <td align="left" style="width: 60%">

        <asp:TextBox ID="TextBox1" runat="server" Width="210px"></asp:TextBox>

      </td>

    </tr>

    <tr>

      <td style="width: 35">
```



```
        Select File

      </td>

      <td style="width: 5%">

        :

      </td>

      <td align="left" style="idth: 60%">

        <asp:FileUpload ID="FileUpload1" runat="server" />

      </td>

    </tr>

    <tr>

      <td style=width: 35%">
```


       Start Date

</td>

<td style="width: 5%">

    :

</td>

<td align="left" style="width: 60%">

   <asp:TextBox ID="TextBox2" runat="server" Width="210px"></asp:TextBox>

</td>

</tr>

<tr>

<td style="width:35%">

       End Date

</td>

<td style="width: 5%">

    :

</td>

<td align="left" style="width: 60%">

   <asp:TextBox ID="TextBox3" runat="server" Width="210px"></asp:TextBox>

</td>

</tr>

<tr>   <td colpan="3">

   <hr style="background-color: #363258; height: 0.2em">

</td>

```
          </tr>


      <tr>

        <td colspan="3">

          <asp:Butto ID="Button1" runat="server" Text="Store" Width="81px"

            onclick="Button1_Click"
/>             &nbs
p;        <asp:Button

              ID="Button3" runat="server" Text="Delete" Width="102px"

            onclick="Button3_Click" />

        </td>

      </tr>

      <tr>

        <td colspan="3">

          <hr style="background-color: #363258; height: 0.2em">

        </td>

      </tr>

      <tr>

        <td colspan="3" align="cnter">

          <asp:GridView ID="GridView1" runat="server" BackColor="White"

            BorderColor="#E7E7FF" BorderStyle="None" BorderWidth="1px" CellPadding="3"
CellSpacing="7"

            GridLines="Horizontal">

            <AlternatingRowStye BackColor="#F7F7F7" />

            <FooterStyle BackColor="#B5C7DE" ForeColor="#4A3C8C" />

            <HeaderStyle BakColor="#4A3C8C" Font-Bold="Tue" ForeColor="#F7F7F7" />

            <PagerStyle BackColor="#E7E7FF" ForeColor="#4A3C8C" HorizontalAlign="Right" />

            <RowStyle BackColor="#E7E7FF" ForeColor="#4A3C8C" />

            <SelectedRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="#F7F7F7" />
```

```
                <SortedAscendingCellSyle BackColor="#F4F4FD" />

                <SortedAscendingHeaderStyle BackColor="#5A4C9D" />

                <SortedDescendingCellStyle BackColor="#D8D80" />

                <SortedDesendingHeaderStyle BackColor="#3E3277" />

            </asp:GridView>

        </td>

      </tr>

    </table>

  </center>

</asp:Content>
```