

UVAS Management System

Project Advisor: Mr. Hafiz Waseem



BSSE 14228

Hafiz Ali Raza

BSSE 14348

Muaz-ul-Hassan

BSSE 14357

Shahbaz Ahmad

BSSE 14360

Atif Amin Janjua

BSSE 14362

Syed M. Ali Shah

UVAS Management System

1 Declaration

This is to certify that this project is submitted by Students of Superior University whose names are mentioned here.

Table 1- Project Participants

Name	Roll No
Hafiz Ali Raza	BSSE-14228
Muaz-ul-Hassan	BSSE-14348
Shahbaz Ahmad	BSSE-14357
Atif Amin	BSSE-14360
Syed Muhammad Ali Shah	BSSE-14362

2 Department Members

Project Manager: Mr. Fahad Sabah



Project Advisor: Mr. Hafiz Waseem





3 Acknowledgement

We wish to thank our parents for their tremendous contribution and support both morally and financially towards the completion of this project.

We are also grateful to our project supervisor Mr. Hafiz Waseem who without his help and guidance this project would not have been completed.

We also show our gratitude towards our all friends and specially my group members who worked side by side in this project.

4 Dedication

The price of success is hard work, dedicated to the job at hand, and the determination that whether we win or lose. We have applied the best of ourselves to the task at hand. It is dedicated to our parents, teachers and our institution. And behind the success of this project is there is a great hand of our parents and our teachers.

Table of Contents

1 Declaration	iii
2 Department Members.....	iv
3 Acknowledgement	v
4 Dedication	v
5 Abstract	2
6 Purpose of the system.....	2
7 Existing system.....	3
8 Problems in the existing system.....	3
9 Risks involved in existing system:.....	3
10 Proposed system	3
11 Scope of the project	4
12 Analysis model.....	5
13 Feasibility study	6
13.1 Technical feasibility	7
13.2 Operational feasibility	7
13.3 Economic feasibility	7
14 Software and Hardware requirements	8
14.1 Environment:	8
14.2 Requirements:.....	9
14.2.1 Hardware requirements:.....	9
14.2.2 Software requirements:	9
15 System Requirements Specification:.....	9
15.1 Developers Responsibilities Overview:	10
15.2 Output Design:	10
16 Introduction to .net framework.....	10
16.1 Principal design features.....	11
16.1.1 Interoperability	11
16.1.2 Common Runtime Engine	11
16.1.3 Base Class Library	11
16.1.4 Simplified Deployment	12

16.1.5 Security	12
16.1.6 Portability	12
16.1.7 Architecture	13
16.1.8 Common Language Infrastructure	13
16.1.9 Assemblies	13
16.1.10 Metadata.....	14
16.1.11 Security	14
16.1.12 Class library	15
16.1.13 Memory management	16
16.2 Active Server Pages.net.....	17
16.3 Language support.....	19
17 Introduction to ASP .NET server controls.....	19
18 C#.NET	20
18.1 Ado.net overview.....	20
18.1.1 Connections.....	22
18.1.2 Commands	22
19 SQL Server	23
19.1 SQL server tables.....	23
19.1.1 Primary key.....	23
19.1.2 Relational database	24
19.1.3 Foreign key.....	24
19.1.4 Referential integrity	24
19.1.5 Data abstraction.....	24
19.1.6 Advantages of RDBMS.....	24
19.1.7 Disadvantages of DBMS.....	25
19.1.8 Features of SQL server (RDBMS).....	25
19.1.9 Enterprise wide data sharing	26
19.1.10 Portability	26
19.1.11 Open systems.....	26
19.1.12 Distributed data sharing	26
19.1.13 Unmatched performance	26
20 Relational Database	26

21 University Management System Project.....	27
22 Categories of University Management System.....	28
22.1 Online web application	28
23 Literature.....	28
24 Content Management System	28
25 Web portal as a Knowledge Management System in the Universities.....	29
26 Student's information.....	29
27 Personal Management Information System	30
28 Accounting.....	31
29 Use cases.....	31
29.1 Use Case for admin.....	31
29.2 Explanation of use case of Admin Panel	32
29.2.1 Login System	32
29.2.2 Add New Admin	32
29.2.3 Add new teacher	32
29.2.4 Add new department	33
29.2.5 View the Student.....	33
29.2.6 View the transport	34
29.2.7 Active and dative transport.....	34
29.2.8 Add new course	34
29.2.9 Add Notification for teacher.....	36
29.2.10 Add Notification for student.....	36
29.3 Use case of teacher	37
29.4 Explanation of use case of teacher Panel	37
29.4.1 Login System	37
29.4.2 Student approval.....	38
29.4.3 Add Lecture	38
29.4.4 Announce Quiz	38
29.4.5 Announce Assignment.....	39
29.4.6 Announce mid term.....	39
29.4.7 Add Timetable.....	39
29.4.8 Calls Attendance	40

29.4.9 Add Results.....	40
29.4.10 Add notification	40
29.5 Use case Of Student	41
29.6 Explanation of use case of Students Panel.....	42
29.6.1 Login System	42
29.6.2 Student Registration	42
29.6.3 Student view lectures	43
29.6.4 Student view timetable.....	43
29.6.5 Student view Results	43
29.6.6 Student view fee.....	44
29.6.7 Student view Attendance.....	44
29.6.8 Student send route	45
29.6.9 Student view Transport.....	45
30 Implementation methodology	45
31 Data Flow Diagram.....	47
31.1 DFD Symbols	48
31.2 Constructing a DFD:.....	49
31.3 Salient features of DFD'S	49
31.4 Types of data flow diagrams	49
31.4.1 Current physical.....	50
31.4.2 Current logical:	50
31.4.3 New logical	50
31.4.4 New physical	50
31.5 Rules governing the DFD'S	50
31.5.1 Process	50
31.5.2 Data store	50
31.5.3 Source or sink	51
31.5.4 Data flow	51
31.6 Admin.....	52
31.6.1 Explanation	52
31.7 Teacher.....	53
31.7.1 Explanation	53

31.8 Student.....	53
31.8.1 Explanation.....	53
32 Class Diagram.....	54
32.1 Admin.....	54
32.2 Student.....	55
33 State Diagrams.....	56
33.1 Admin.....	56
33.2 Student.....	57
33.3 Teacher.....	58
34 Database Tables.....	59
34.1 Users	59
34.1.1 Description.....	59
34.2 Departments	59
34.2.1 Description.....	60
34.3 Course.....	60
34.3.1 Description.....	60
34.4 Student Info	60
34.4.1 Description.....	61
34.5 Student Courses	61
34.5.1 Description.....	62
34.6 Results.....	62
34.6.1 Description.....	62
34.7 Fee Info	63
34.7.1 Description.....	63
35 Testing	63
35.1 Strategic approach to software testing.....	64
35.2 Unit testing.....	65
35.3 White box testing	65
35.4 Basic path testing	66
35.6 Conditional testing.....	66
35.7 Data flow testing.....	66
35.8 Loop testing	67

36 User Interfaces	67
36.1 User Interface Design:	67
36.2 User Interface Systems can be broadly classified as:	67
36.3 Login	67
36.3.1 Description:	68
36.4 Admin	68
36.4.1 Add courses	68
36.4.2 Send Notification	69
36.4.3 Add New Admin	69
36.4.4 Dashboard	70
36.4.5 Add new Teacher	72
36.4.6 View Students	73
36.5 Teacher	73
36.5.1 Add Result	74
36.5.2 Send Notification	76
36.5.3 Add Time Table	77
36.5.4 Add Attendance	78
36.5.5 Dashboard	78
36.5.6 View Notification	80
36.5.7 Time able	80
36.6 Student	80
36.6.1 View/Download Results	81
.....	81
36.6.3 Transport	82
37 System Security	82
37.1 Security in software	83
37.1.1 Client side validation	83
37.1.2 Server side validation	84
38 Maintenance	84
39 Conclusion	85
40 Limitation	86
41 Future Work	86

42 References	88
---------------------	----

5 Abstract

Record updating is main issue in universities in Pakistan and a much more time is required to update records in desktop applications. In this project a web based system has been developed for UVAS (University of Veterinary and Animal Sciences). This online system will replace the manual desktop application system currently used in common University. This system will contribute in minimizing the time required in storing, processing, retrieving and organizing data in the organization. More over the system will help in improving communication among the number of each department. This system is called University Management System.

Every origination weather big or small has challenges to overcome and managing the information of the students, faculty registration courses every university has different needs therefor we design this type of management system according to UVAS management system. These system will allow better management.

We have introduced online registration system and online access to individual students as well as to the Teachers of the University. Student can get himself/herself not only registered but can also see his progressive report too. Result will be shown to individual panel and many more.

6 Purpose of the system

UNIVERSITY MANAGEMENT SYSTEM [UMS] deals with the maintenance of university, college, faculty, student information within the university. This project of UMS involved the automation of student information that can be implemented in different college managements.

The project deals with retrieval of information through an INTRANET based campus wide portal. It collects related information from all the departments of an organization and maintains files, which are used to generate reports in various forms to measure individual and overall performance of the students.

7 Existing system

The system starts with registration of new staff and students. When the subjects are to be allocated to the faculty, the Head of the Department should enter everything in the Excel sheets. Then the staff enters corresponding subject's attendance and marks of a student then those must also be entered in the Excel sheets and validations are to be done by the user itself. So there will be a lot of work to be done and must be more conscious during the entrance of details. So, more risk is involved.

8 Problems in the existing system

Storing and accessing the data in the form of Excel sheets and account books is a tedious work. It requires a lot of laborious work. It may often yield undesired results. Maintaining these records as piles may turn out to be a costlier task than any other of the colleges and institutions

9 Risks involved in existing system:

Present System is time-consuming and also results in lack of getting inefficient results.

Some of the risks involved in the present system are:

- During the entrance of marks and attendance, if any mistake is done at a point, then this becomes cumulative and leads to adverse consequences
- If there is any need to retrieve results it may seem to be difficult to search.

10 Proposed system

UMS (UNIVERSITY MANAGEMENT SYSTEM) makes management to get the most updated information always by avoiding manual accounting process. This system has the following functional divisions.

- University Administrator

- College Administrator
- User (Students / Faculties)

University Administrator has the functionality of registering new colleges and courses. **College Administrator** has the rights of creating department, allocating courses to departments, creating faculties, students and allocating subjects to faculties, and modifications in the data entered by the user can also be done by the college administrator.

User of this may be faculty or students. Faculty has the facility of entering the marks and attendance of the students. Students can check their marks and attendance but there is no chance of modifications.

Reports must be generated for the existing data i.e. for attendance and marks of the students, which are used to assess the performance of the students. These reports should be viewed by the in charge and user.

11Scope of the project

It may help perfect management in detail. It will help to reduce the cost of management and collection procedure will go on smoothly.

Our project's aim is to business process automation we have tried to computerize various processes of university.

- In computer system the persons has to fill the various forms and numbers of copies. The forms can be easily generated at a time.
- In computer system it is not necessary to create a manifest but we can directly print it which saves our time.
- To utilize resources in efficient manner by increasing their productivity through automation.
- Be easy to understand by the user and operator.
- Easy to operate.
- Have a good user interface

- Delivered on schedule on budget.
- Be expandable.

12 Analysis model

The model that is basically being followed is the WATER FALL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study is done. Once that part is over the requirement analysis and project planning begins. The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are:

- Requirement Analysis
- Project Planning
- System design
- Detail design
- Coding
- Unit testing
- System integration & testing

Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of other phase. The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of spiral model are also incorporated like after the people concerned with the project review completion of each of the phase the work done.

WATER FALL MODEL was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

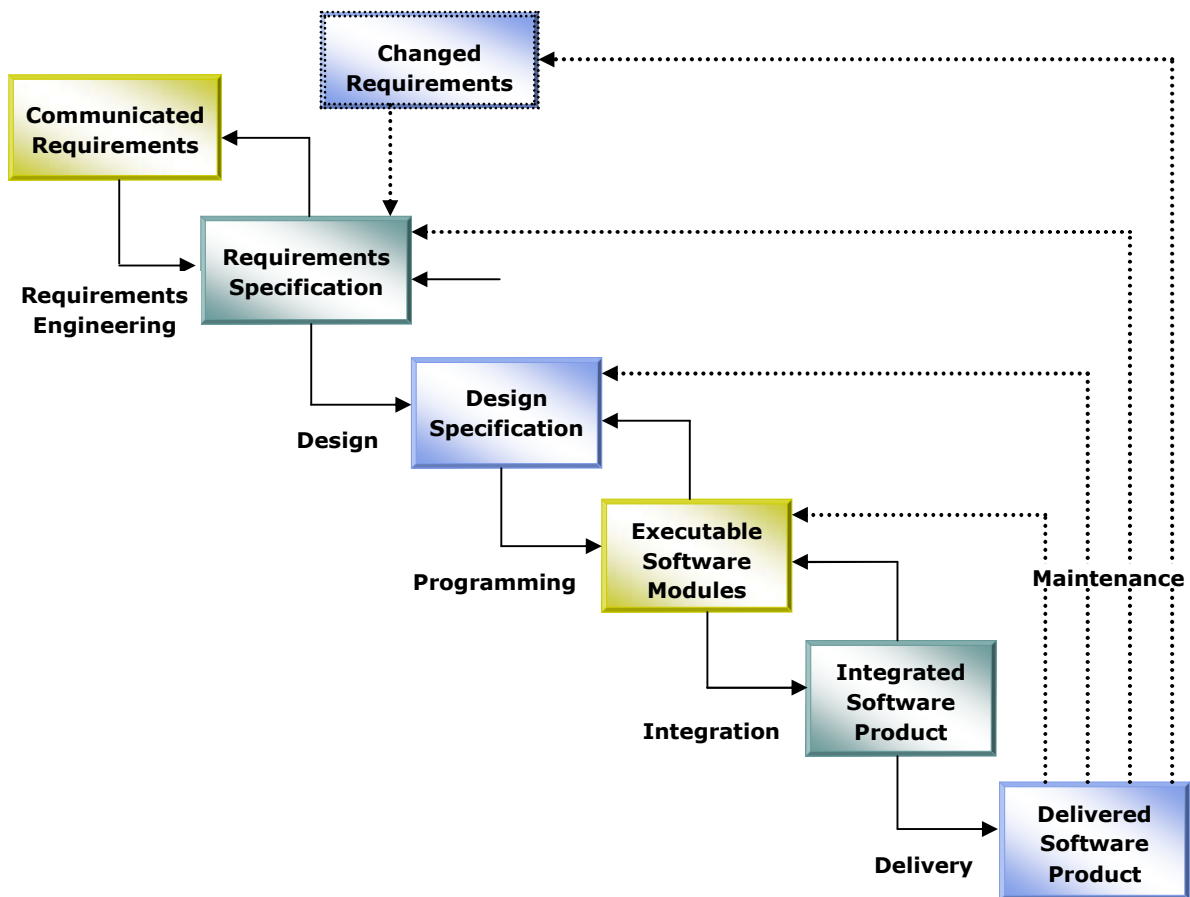


Figure 1- Water Fall Model

13 Feasibility study

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- **Technical Feasibility**
- **Operational Feasibility**
- **Economic Feasibility**

13.1 Technical feasibility

Technical Feasibility centers on the existing computer system hardware, software, etc. and to some extent how it can support the proposed addition. This involves financial considerations to accommodate technical enhancements. Technical support is also a reason for the success of the project. The techniques needed for the system should be available and it must be reasonable to use. Technical Feasibility is mainly concerned with the study of function, performance, and constraints that may affect the ability to achieve the system. By conducting an efficient technical feasibility we need to ensure that the project works to solve the existing problem area.

Since the project is designed with ASP.NET with C# as Front end and SQL Server 2000 as Back end, it is easy to install in all the systems wherever needed. It is more efficient, easy and user-friendly to understand by almost everyone. Huge amount of data can be handled efficiently using SQL Server as back end. Hence this project has good technical feasibility

13.2 Operational feasibility

People are inherently instant to change and computers have been known to facilitate change. An estimate should be made to how strong a reaction the user staff is likely to have towards the development of the computerized system.

The staff is accustomed to computerized systems. These kinds of systems are becoming more common day by day for evaluation of the software engineers. Hence, this system is operationally feasible. As this system is technically, economically and operationally feasible, this system is judged feasible.

13.3 Economic feasibility

The role of interface design is to reconcile the differences that prevail among the software engineer's design model, the designed system meet the end user requirement with economical way at minimal cost within the affordable price by encouraging more of

proposed system. Economic feasibility is concerned with comparing the development cost with the income/benefit derived from the developed system. In this we need to derive how this project will help the management to take effective decisions.

Economic Feasibility is mainly concerned with the cost incurred in the implementation of the software. Since this project is developed using ASP.NET with C# and SQL Server which is more commonly available and even the cost involved in the installation process is not high.

Similarly it is easy to recruit persons for operating the software since almost all the people are aware of ASP.NET with C# and SQL Server. Even if we want to train the persons in these area the cost involved in training is also very less. Hence this project has good economic feasibility.

The system once developed must be used efficiently. Otherwise there is no meaning for developing the system. For this a careful study of the existing system and its drawbacks are needed. The user should be able to distinguish the existing one and proposed one, so that one must be able to appreciate the characteristics of the proposed system, the manual one is not highly reliable and also is considerably fast. The proposed system is efficient, reliable and also quickly responding.

14 Software and Hardware requirements

14.1 Environment:

- **Servers:**
 - **Operating System Server:** - Microsoft Windows 2000 or Higher
 - **Data Base Server:** Microsoft SQL Server 2000/2005
- **Clients** : Microsoft Internet Explorer,
- **Tools** : Microsoft Visual Studio .Net
- **User Interface:** ASP.NET with AJAX
- **Code Behind** : VC#.NET

14.2 Requirements:

14.2.1 Hardware requirements:

Number	Description
1	PC with 2 GB hard-disk and 256 MB RAM

14.2.2 Software requirements:

Number	Description
1	Windows 2000/ XP/ or Higher with MS-office
2	MS-SQL server2000/2005
3	Ms-Visual Studio .Net 2005
4	Ms-Internet Explorer

15 System Requirements Specification:

The software, Site Explorer is designed for management of web sites from a remote location.

Purpose: The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

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

15.1 Developers Responsibilities Overview:

The developer is responsible for:

- Developing the system, which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

15.2 Output Design:

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization.
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

16 Introduction to .net framework

The Microsoft .NET Framework is a software technology that is available with several Microsoft Windows operating systems. It includes a large library of pre-coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The .NET Framework is a key Microsoft offering and is intended to be used by most new applications created for the Windows platform.

The pre-coded solutions that form the framework's Base Class Library cover a large range of programming needs in a number of areas, including user interface, 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 compose the .NET Framework.

16.1 Principal design features

16.1.1 Interoperability

Because interaction between new and older applications is commonly required, the .NET Framework provides means to access functionality that is implemented in programs that execute outside the .NET environment. Access to COM components is provided in the `System.Runtime.InteropServices` and `System.EnterpriseServices` namespaces of the framework; access to other functionality is provided using the `P/Invoke` feature.

16.1.2 Common 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.

16.1.3 Base Class Library

The Base Class Library (BCL), part of the Framework Class Library (FCL), is a library of functionality available to all languages 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.

16.1.4 Simplified Deployment

Installation of computer software must be carefully managed to ensure that it does not interfere with previously installed software, and that it conforms to security requirements. The .NET framework includes design features and tools that help address these requirements.

16.1.5 Security

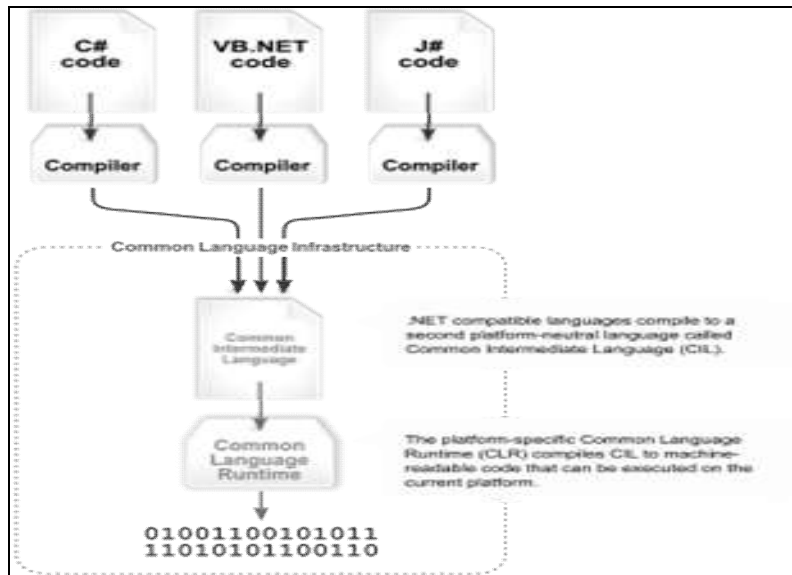
The design is meant to address some of the vulnerabilities, such as buffer overflows, that have been exploited by malicious software. Additionally, .NET provides a common security model for all applications.

16.1.6 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.

In addition, Microsoft submits the specifications for the Common Language Infrastructure (which includes the core class libraries, Common Type System, and the Common Intermediate Language), the C# language, and the C++/CLI language 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.

16.1.7 Architecture



Visual overview of the Common Language Infrastructure (CLI)

16.1.8 Common Language Infrastructure

The core aspects of the .NET framework lie within the Common Language Infrastructure, or CLI. The purpose of the CLI is to provide a language-neutral platform for application development and execution, including functions for exception handling, garbage collection, security, and interoperability. Microsoft's implementation of the CLI is called the Common Language Runtime or CLR.

16.1.9 Assemblies

The intermediate CIL code is housed in .NET assemblies. As mandated by specification, assemblies are stored in the Portable Executable (PE) format, common on the Windows platform for all DLL and EXE files. The assembly consists of one or more files, one of which must contain the manifest, which has the metadata for the assembly. The complete name of an assembly (not to be confused with the filename on disk) contains its simple text name, version number, culture, and public key token. The public key token is a unique hash generated when the assembly is compiled, thus two assemblies with the same public key token are guaranteed to be identical from the point of view of the framework. A private key can also be specified known only to the creator of the assembly and can be used for strong naming and to guarantee that the assembly is from

the same author when a new version of the assembly is compiled (required to add an assembly to the Global Assembly Cache).

16.1.10 Metadata

All CLI is self-describing through .NET metadata. The CLR checks the metadata to ensure that the correct method is called. Metadata is usually generated by language compilers but developers can create their own metadata through custom attributes. Metadata contains information about the assembly, and is also used to implement the reflective programming capabilities of .NET Framework.

16.1.11 Security

.NET has its own security mechanism with two general features: Code Access Security (CAS), and validation and verification. Code Access Security is based on evidence that is associated with a specific assembly. Typically the evidence is the source of the assembly (whether it is installed on the local machine or has been downloaded from the intranet or Internet).

Code Access Security uses evidence to determine the permissions granted to the code. Other code can demand that calling code is granted a specified permission.

The demand causes the CLR to perform a call stack walk: every assembly of each method in the call stack is checked for the required permission; if any assembly is not granted the permission a security exception is thrown.

When an assembly is loaded the CLR performs various tests. Two such tests are validation and verification. During validation the CLR checks that the assembly contains valid metadata and CIL, and whether the internal tables are correct. Verification is not so exact. The verification mechanism checks to see if the code does anything that is 'unsafe'. The algorithm used is quite conservative; hence occasionally code that is 'safe' does not pass. Unsafe code will only be executed if the assembly has the 'skip verification' permission, which generally means code that is installed on the local machine.

.NET Framework uses app-domains as a mechanism for isolating code running in a process. App-domains can be created and code loaded into or unloaded from them independent of other app-domains. This helps increase the fault tolerance of the application, as faults or crashes in one app-domain do not affect rest of the application.

App-domains can also be configured independently with different security privileges. This can help increase the security of the application by isolating potentially unsafe code. The developer, however, has to split the application into sub domains; it is not done by the CLR.

16.1.12 Class library

Namespaces in the BCL
System
System. CodeDom
System. Collections
System. Diagnostics
System. Globalization
System. IO
System. Resources
System. Text
System.Text.RegularExpressions

Microsoft .NET Framework includes a set of standard class libraries. The class library is organized in a hierarchy of namespaces. Most of the built in APIs are part of either System. or Microsoft.Namespaces. It encapsulates a large number of common functions, such as file reading and writing, graphic rendering, database interaction, and XML document manipulation, among others. The .NET class libraries are available to all .NET languages. The .NET Framework class library is divided into two parts: the Base Class Library and the Framework Class Library.

The Base Class Library (BCL) includes a small subset of the entire class library and is the core set of classes that serve as the basic API of the Common Language

Runtime. The classes in `mscorlib.dll` and some of the classes in `System.dll` and `System.core.dll` are considered to be a part of the BCL. The BCL classes are available in both .NET Framework as well as its alternative implementations including .NET Compact Framework, Microsoft Silver light and Mono.

The Framework Class Library (FCL) is a superset of the BCL classes and refers to the entire class library that ships with .NET Framework. It includes an expanded set of libraries, including Win Forms, ADO.NET, ASP.NET, Language Integrated Query, Windows Presentation Foundation, Windows Communication Foundation among others. The FCL is much larger in scope than standard libraries for languages like C++, and comparable in scope to the standard libraries of Java.

16.1.13 Memory management

The .NET Framework CLR frees the developer from the burden of managing memory (allocating and freeing up when done); instead it does the memory management itself. To this end, the memory allocated to instantiations of .NET types (objects) is done contiguously from the managed heap, a pool of memory managed by the CLR. As long as there exists a reference to an object, which might be either a direct reference to an object or via a graph of objects, the object is considered to be in use by the CLR. When there is no reference to an object, and it cannot be reached or used, it becomes garbage. However, it still holds on to the memory allocated to it. .NET Framework includes a garbage collector which runs periodically, on a separate thread from the application's thread, that enumerates all the unusable objects and reclaims the memory allocated to them.

The .NET Garbage Collector (GC) is a non-deterministic, compacting, and markswEEP garbage collector. The GC runs only when a certain amount of memory has been used or there is enough pressure for memory on the system. Since it is not guaranteed when the conditions to reclaim memory are reached, the GC runs are non-deterministic. Each .NET application has a set of roots, which are pointers to objects on the managed heap (managed objects). These include references to static objects and objects defined as local variables or method parameters currently in scope, as well as objects referred to by CPU registers. When the GC runs, it pauses the application, and for each object referred to in the root, it recursively enumerates all the objects reachable from the root objects and marks them as reachable.

It uses .NET metadata and reflection to discover the objects encapsulated by an object, and then recursively walk them. It then enumerates all the objects on the heap (which were initially allocated contiguously) using reflection. All objects not marked as reachable are garbage. This is the mark phase. Since the memory held by garbage is not of any consequence, it is considered free space. However, this leaves chunks of free space between objects which were initially contiguous. The objects are then compacted together, by using memory to copy them over to the free space to make them contiguous again. Any reference to an object invalidated by moving the object is updated to reflect the new location by the GC. The application is resumed after the garbage collection is over.

The GC used by .NET Framework is actually generational. Objects are assigned a generation; newly created objects belong to Generation 0. The objects that survive a garbage collection are tagged as Generation 1, and the Generation 1 objects that survive another collection are Generation 2 objects. The .NET Framework uses up to Generation 2 objects. Higher generation objects are garbage collected less frequently than lower generation objects. This helps increase the efficiency of garbage collection, as older objects tend to have a larger lifetime than newer objects. Thus, by removing older (and thus more likely to survive a collection) objects from the scope of a collection run, fewer objects need to be checked and compacted.

Versions: Microsoft started development on the .NET Framework in the late 1990s originally under the name of Next Generation Windows Services (NGWS). By late 2000 the first beta versions of .NET 1.0 were released.

16.2 Active Server Pages.net

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

Enhanced Performance. ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.

World-Class Tool Support. The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

Power and Flexibility. Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.

Simplicity. ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

Manageability. ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

Scalability and Availability. ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

Customizability and Extensibility. ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.

Security. With built in Windows authentication and per-application configuration, you can be assured that your applications are secure.

16.3 Language support

The Microsoft .NET Platform currently offers built-in support for three languages: C#, Visual Basic, and Java Script.

17 Introduction to ASP.NET server controls

In addition to (or instead of) using `<% %>` code blocks to program dynamic content, ASP.NET page developers can use ASP.NET server controls to program Web pages. Server controls are declared within an .aspx file using custom tags or intrinsic HTML tags that contain a `runat="server"` attributes value. Intrinsic HTML tags are handled by one of the controls in the `System.Web.UI.HtmlControls` namespace. Any tag that doesn't explicitly map to one of the controls is assigned the type of `System.Web.UI.HtmlControls.HtmlGenericControl`.

Server controls automatically maintain any client-entered values between round trips to the server. This control state is not stored on the server (it is instead stored within

an `<input type="hidden">` form field that is round-tripped between requests). Note also that no client-side script is required.

In addition to supporting standard HTML input controls, ASP.NET enables developers to utilize richer custom controls on their pages. For example, the following sample demonstrates how the `<asp:adrotator>` control can be used to dynamically display rotating ads on a page.

1. ASP.NET Web Forms provide an easy and powerful way to build dynamic Web UI.
2. ASP.NET Web Forms pages can target any browser client (there are no script library or cookie requirements).
3. ASP.NET Web Forms pages provide syntax compatibility with existing ASP pages.
4. ASP.NET server controls provide an easy way to encapsulate common functionality.
5. ASP.NET ships with 45 built-in server controls. Developers can also use controls built by third parties.
6. ASP.NET server controls can automatically project both up-level and down-level HTML.
7. ASP.NET templates provide an easy way to customize the look and feel of list server controls.
8. ASP.NET validation controls provide an easy way to do declarative client or server data validation.

18 C#.NET

18.1 ADO.NET overview

ADO.NET is an evolution of the ADO data access model that directly addresses user requirements for developing scalable applications. It was designed specifically for the web with scalability, statelessness, and XML in mind.

ADO.NET uses some ADO objects, such as the Connection and Command objects, and also introduces new objects. Key new ADO.NET objects include the DataSet, Data Reader, and Data Adapter.

The important distinction between this evolved stage of ADO.NET and previous data architectures is that there exists an object -- the DataSet -- that is separate and distinct from any data stores. Because of that, the DataSet functions as a standalone entity. You can think of the DataSet as an always disconnected recordset that knows nothing about the source or destination of the data it contains. Inside a DataSet, much like in a database, there are tables, columns, relationships, constraints, views, and so forth.

A DataAdapter is the object that connects to the database to fill the DataSet. Then, it connects back to the database to update the data there, based on operations performed while the DataSet held the data. In the past, data processing has been primarily connection-based. Now, in an effort to make multi-tiered apps more efficient, data processing is turning to a message-based approach that revolves around chunks of information. At the center of this approach is the DataAdapter, which provides a bridge to retrieve and save data between a DataSet and its source data store. It accomplishes this by means of requests to the appropriate SQL commands made against the data store.

The XML-based DataSet object provides a consistent programming model that works with all models of data storage: flat, relational, and hierarchical. It does this by having no 'knowledge' of the source of its data, and by representing the data that it holds as collections and data types. No matter what the source of the data within the DataSet is, it is manipulated through the same set of standard APIs exposed through the DataSet and its subordinate objects.

While the DataSet has no knowledge of the source of its data, the managed provider has detailed and specific information. The role of the managed provider is to connect, fill, and persist the DataSet to and from data stores. The OLE DB and SQL Server .NET Data Providers (System.Data.OleDb and System.Data.SqlClient) that are

part of the .Net Framework provide four basic objects: the Command, Connection, DataReader and DataAdapter. In the remaining sections of this document, we'll walk through each part of the DataSet and the OLE DB/SQL Server .NET Data Providers explaining what they are, and how to program against them.

The following sections will introduce you to some objects that have evolved, and some that are new. These objects are:

- Connections. For connection to and managing transactions against a database.
- Commands. For issuing SQL commands against a database.
- DataReaders. For reading a forward-only stream of data records from a SQL Server data source.
- DataSet. For storing, Remoting and programming against flat data, XML data and relational data.
- DataAdapters. For pushing data into a DataSet, and reconciling data against a database.

When dealing with connections to a database, there are two different options: SQL Server .NET Data Provider (System.Data.SqlClient) and OLE DB .NET Data Provider (System.Data.OleDb). In these samples we will use the SQL Server .NET Data Provider. These are written to talk directly to Microsoft SQL Server. The OLE DB .NET Data Provider is used to talk to any OLE DB provider (as it uses OLE DB underneath).

18.1.1 Connections

Connections are used to 'talk to' databases, and are represented by provider-specific classes such as SqlConnection. Commands travel over connections and result sets are returned in the form of streams which can be read by a DataReader object, or pushed into a DataSet object.

18.1.2 Commands

Commands contain the information that is submitted to a database, and are represented by provider-specific classes such as Sql-Command. A command can be a stored procedure call, an UPDATE statement, or a statement that returns results. You can also use input and output parameters, and return values as part of your command syntax.

The example below shows how to issue an INSERT statement against the Northwind database.

19 SQLServer

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

19.1 SQL server tables

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

19.1.1 Primary key

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

19.1.2 Relational database

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the tables and enables you to define relationships between the tables.

19.1.3 Foreign key

When a field in one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table

19.1.4 Referential integrity

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

19.1.5 Data abstraction

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. Data abstraction is divided into three levels.

Physical level: This is the lowest level of abstraction at which one describes how the data are actually stored.

Conceptual Level: At this level of database abstraction all the attributes and what data are actually stored is described and entries and relationship among them.

View level: This is the highest level of abstraction at which one describes only part of the database.

19.1.6 Advantages of RDBMS

- Redundancy can be avoided
- Inconsistency can be eliminated
- Data can be Shared

- Standards can be enforced
- Security restrictions can be applied
- Integrity can be maintained
- Conflicting requirements can be balanced
- Data independence can be achieved.

19.1.7 Disadvantages of DBMS

A significant disadvantage of the DBMS system is cost. In addition to the cost of purchasing or developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage.

While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in case of failure the data can be recovered.

19.1.8 Features of SQL server (RDBMS)

SQL SERVER is one of the leading database management systems (DBMS) because it is the only Database that meets the uncompromising requirements of today's most demanding information systems. From complex decision support systems (DSS) to the most rigorous online transaction processing (OLTP) application, even application that require simultaneous DSS and OLTP access to the same critical data, SQL Server leads the industry in both performance and capability.

SQL SERVER is a truly portable, distributed, and open DBMS that delivers unmatched performance, continuous operation and support for every database.

SQL SERVER RDBMS is high performance fault tolerant DBMS which is specially designed for online transactions processing and for handling large database application.

SQL SERVER with transactions processing option offers two features which contribute to very high level of transaction processing throughput, which are

- The row level lock manager

19.1.9 Enterprise wide data sharing

The unrivaled portability and connectivity of the SQL SERVER DBMS enables all the systems in the organization to be linked into a singular, integrated computing resource

19.1.10 Portability

SQL SERVER is fully portable to more than 80 distinct hardware and operating systems platforms, including UNIX, MSDOS, OS/2, Macintosh and dozens of proprietary platforms. This portability gives complete freedom to choose the database server platform that meets the system requirements.

19.1.11 Open systems

SQL SERVER offers a leading implementation of industry-standard SQL. SQL Server's open architecture integrates SQL SERVER and non –SQL SERVER DBMS with industry's most comprehensive collection of tools, application, and third party software products SQL Server's Open architecture provides transparent access to data from other relational database and even non-relational database.

19.1.12 Distributed data sharing

SQL Server's networking and distributed database capabilities to access data stored on remote server with the same ease as if the information was stored on a single local computer. A single SQL statement can access data at multiple sites. You can store data where system requirements such as performance, security or availability dictate.

19.1.13 Unmatched performance

The most advanced architecture in the industry allows the SQL SERVER DBMS to deliver unmatched performance.

20Relational Database

SQL server[1] is used to develop database. Relational tables ensure that the data they contain stay accurate and are always accessible. First, the rows in a relational table should all be physically separate. If there is a duplicate row, there can be problems resolving which of two possible selections the correct one is. For most DBMSs, the user

can specify that duplicate row is not allowed, and if that is done, the DBMS will prevent the addition of any rows that duplicates an existing row.

Another integrity rule of the traditional relational model is that column values must not be repeating groups or arrays. A third aspect of data integrity involves the concept of null values. A database takes care of situations where data may not be available by using null values to indicate that a value is missing. It does not equate to a blank or zero. A blank is considered equal to another blank, a zero is equal to another zero, but two null values are not considered to be equal.

When each row in a table is different, it is possible to use one or more columns to identify a particular row. This unique column or group of columns is called a primary key. Any column that is part of a primary key cannot be null if it were, the primary key containing it would no longer be a complete identifier. This rule is referred to as entity integrity.

The teachers table provides some of these relational database concepts. It has five columns and six rows, with each row representing a different Teacher.

21University Management System Project

University Management System[2] is a project having back-end SQL server. In **University Management System** enroll all kinds of branches which are available in University also can take to enter faculty members detailed information related to each and every branch, detailed information like personal information, salary details of staff members, leave details, etc. Administrator of **University Management System Project** has sufficient right to enter student information system, fee structure, course details and other information. Actually University information system project has two types of users, admin and sub user categories available. In these categories varieties sub categories stored to maintain information.

22Categories of University Management System

22.1 Online web application

For Brief Info and implementation process, or some basic idea of University management system you can watch this project video to get more detail info of University management system project in asp.net source code.

University management system Requirement

- Font End Application: ASP.Net
- Back End Software: SQL Database
- System Requirement: WindowsXP, 7, 8.1, 10
- Web browser: Google Chrome or any other

23Literature

- University management system project is a web application which is implemented in Asp .net MVC platform[3].
- University management system project tutorial and guide for developing code.
- Data flow diagram(DFD)
- Sequence diagram and software requirements specification (SRS) of Universitymanagement system in report file.
- Data Communication and Networking applied on system layers

24Content Management System

University uses a **Content Management System (CMS)** named to make it easy for the University's divisions and departments to manage their own web presence and stay online to access information and to respond their students regarding their answers.

Platform & Integration Solutions actively works with staff, faculty, and students to help them create and manage their website's content. As a result, users have all the tools to publish web pages and articles to kent.edu easily and efficiently for the world to see.

Teacher can update student record like papers, quizzes and assignment marks which can be viewed by related student. Student can request to edit his/her record but he/she could not edit the record.

25Web portal as a Knowledge Management System in the Universities

Web portals[4] become important because of student's need to access the required information online. It is vital that universities have a dynamic connection with the students by sharing the organized knowledge via the portal and help the students from messy information on the Web.

There are various types of web portals with various utilities that provide benefits to the users. Each user can have his/her own distinctive definition of his/her web portals. Simply, a portal is a gateway to online network accessible resources through the Intranet, extranet or Internet, thus a simple web page could adjust this definition, as could a complex site involved of thousands of web pages.

The early portals were very simple, offering their members a static view of contents from a small number of sources. Generally, a web portal allows the users to access information from varied sources in an integrated way. Apart from the standard search engine, web portals provide other services such as e-mail, news, stock prices, information, and entertainment, depending on the nature of business of the portals host institutions.

26Student's information

Student Information Management System (SIMS) provides a simple interface for maintaining the student information. It can be used by educational institutes or Universities to maintain the records of students easily and accurately. The creation and management of accurate, up-to-date information regarding a students' academic career is critically important in the University as well as Universities.

Student information system deals with all kind of student details, academic related reports, University details, course details, curriculum, batch details, placement details and other resource related details too. It tracks all the details of a student from the day one to the end of the course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed years, coming year curriculum details, exam details, project or any other assignment details, final exam result and all these will be available through a secure, online interface embedded in the University's website.

It will also have faculty details, batch execution details, students' details in all aspects, the various academic notifications to the staff and students updated by the University administration. It also facilitate us explore all the activities happening in the University. Different reports and Queries can be generated based on vast options related to students, batch, course, faculty, examsand certification even for the entire University.

27Personal Management Information System

This module will keep details pertaining

- To each Teacher with regard to his/her service details which will include salary, leave, training and academic records.
- To each administrator with regard to his/her service details which will include salary, leave, training and academic records.
- To each Student with regard to his/her educational details which will include personal details, leave, course and academic records.

28Accounting

In this section we will computerize voucher/bill.

- This includes Finance and Accounts
 - Expenditure done by University
 - Collection from fees, securities, non-refundable/refundable funds.

29Use cases

Use cases [5]

29.1 Use Case for admin

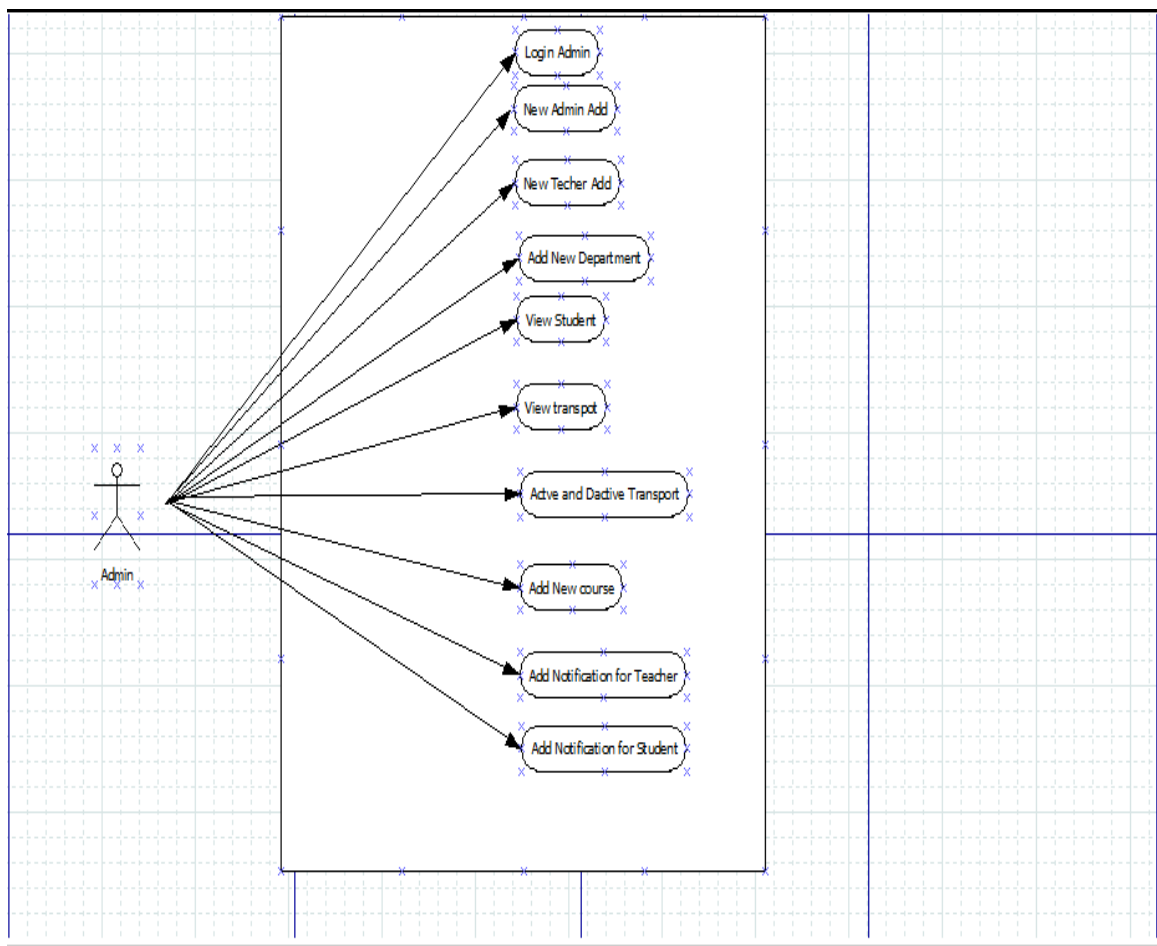


Figure 2

29.2 Explanation of use case of Admin Panel

29.2.1 Login System

Description :	This use case describes the login system after login admin can manage the system.
Actor :	Admin
Precondition :	All actors must have the sign in identification 1. The actor wants to use the application. 2. Click on 'Sign In' link on top of the homepage.
Scenarios	a. If Admin doesn't have an account, they must sign up. 3. Actor sign in to the application. 4. The authenticate system will validate the student id.
Post Conditions:	None

29.2.2 Add New Admin

Description :	This use case describes the Admin can add New Admin in this system
Actor :	Admin
Precondition :	All actors must have the sign in identification 1. The actor wants to use the application. 2. Click on 'Sign In' link on top of the homepage.
Scenarios	a. If Admin doesn't have an account, they must sign up. 3. Actor signs in to the application. 4. The authenticate system will validate the student id.
Post Conditions:	None

29.2.3 Add new teacher

Description :	This use case describes the Admin can add New teacher in this system
Actor :	

Admin

Precondition :

All actors must have the sign in identification
System shall allow to admin view profile of all members
employee view to their and student profile. Students can
also view to their own profile in online university
management system.

Scenarios

Post Conditions:

None

29.2.4 Add new department

Description :

This use case describes the Admin can add New department in this system

Actor :

Admin

Precondition :

All actors must have the sign in identification
System shall allow to admin can add the new department
if university must study this department. Students can
also view to their own profile in online university
management system.

Scenarios

Post Conditions:

None

29.2.5 View the Student

Description :

This use case describes the Admin can view the student in this system

Actor :

Admin

Precondition :

All actors must have the sign in identification
System shall allow to admin can view the student if student
can register himself or herself in the university management
system.

Scenarios

Post Conditions: None

29.2.6 View the transport

Description : This use case describes the Admin can view the transport of the student in this system

Actor : Admin

Precondition : All actors must have the sign in identification
System shall allow to admin can view the transport if student can send himself or herself route in the university management system.

Scenarios

Post Conditions: None

29.2.7Active and dative transport

Description : This use case describes the Admin can active and d active the student route in this system

Actor : Admin

Precondition : All actors must have the sign in identification
System shall allow to admin can active and d active the student route if student can send himself or herself route in the university management system.

Scenarios

Post Conditions: None

29.2.8Add new course

Description : This use case describes the Admin can add new course in this system

Actor : Admin

Precondition : All actors must have the sign in identification
System shall allow to admin can add the new course if university studying new course in the

Scenarios

university management system.

Post Conditions:

None

29.2.9 Add Notification for teacher

Description :	This use case describes the Admin can add notification for teacher in this system
Actor :	Admin
Precondition :	All actors must have the sign in identification System shall allow to admin can add notification for teacher if admin can send to notification in the university management system.
Scenarios	
Post Conditions:	None

29.2.10 Add Notification for student

Description :	This use case describes the Admin can add notification for student in this system
Actor :	Admin
Precondition :	All actors must have the sign in identification System shall allow to admin can add notification for student if admin can send to notification in the university management system.
Scenarios	
Post Conditions:	None

29.3 Use case of teacher

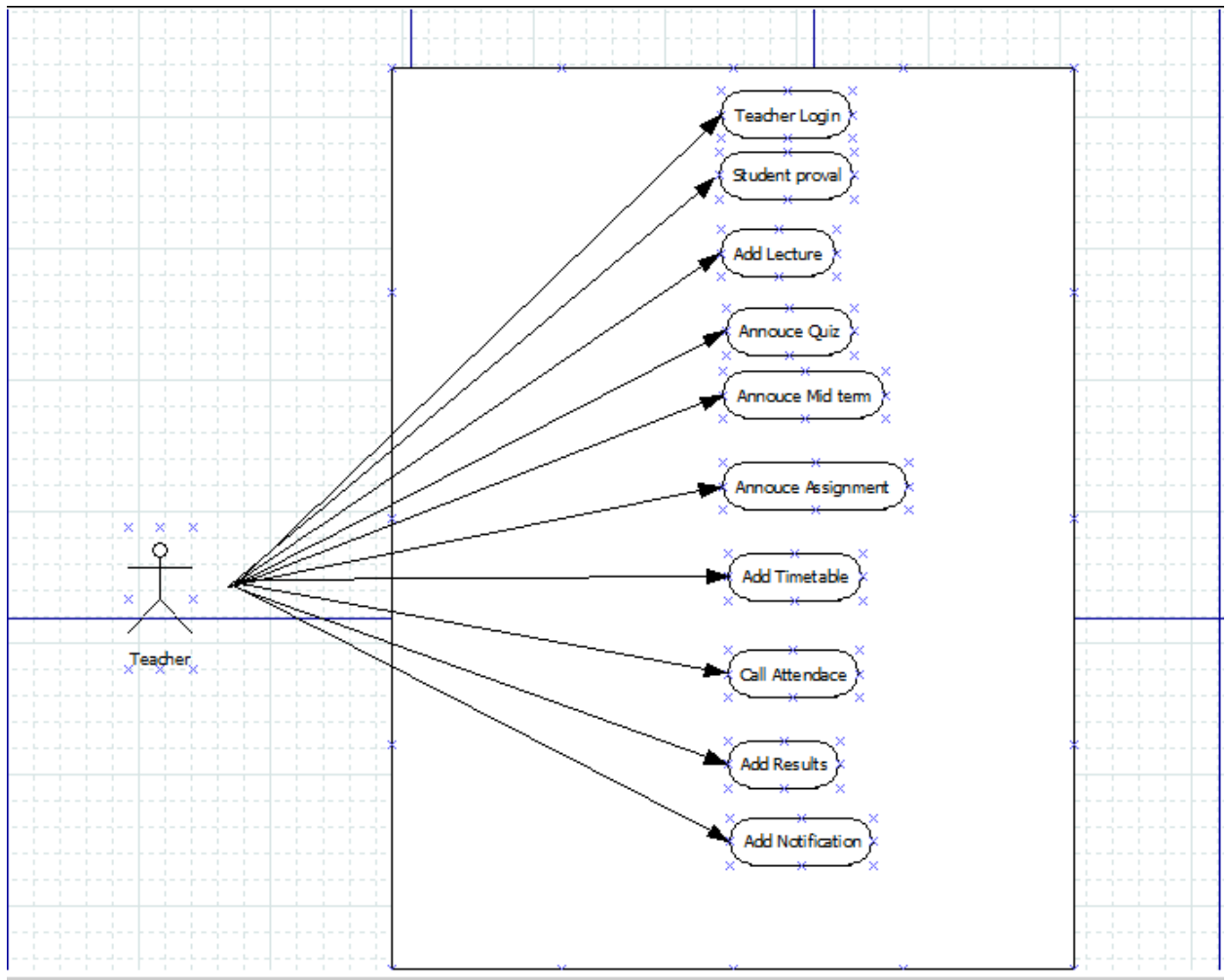


Figure 3

29.4 Explanation of use case of teacher Panel

29.4.1 Login System

Description : This use case describes the login system after login Teacher can manage the system.

Actor : Teacher

Precondition : All actors must have the sign in identification

Scenarios

1. The actor wants to use the application.
2. Click on 'Sign In' link on top of the homepage.

- a. If teacher doesn't have an account, they must sign up.
- 3. Actor sign in to the application.
- 4. The authenticate system will validate the student id.

Post Conditions: None

29.4.2 Student approval

Description :	This use case describes the teacher can approval the student registration in this system
Actor :	Teacher
Precondition :	All actors must have the sign in identification System shall allow to Teacher can approval the student registration if student can register in the university management system.
Scenarios	
Post Conditions:	None

29.4.3 Add Lecture

Description :	This use case describes the teacher can add the lecture this system
Actor :	Teacher
Precondition :	All actors must have the sign in identification System shall allow to Teacher can add the lecture which can take the lecture in the university management system.
Scenarios	
Post Conditions:	None

29.4.4 Announce Quiz

Description :	This use case describes the teacher can announce quiz this system
Actor :	Teacher

Precondition : All actors must have the sign in identification
System shall allow to Teacher can announce the quiz if teacher want to this in the university management system.

Scenarios

Post Conditions: None

29.4.5 Announce Assignment

Description : This use case describes the teacher can announce assignment this system

Actor : Teacher

Precondition : All actors must have the sign in identification
System shall allow to Teacher can announce the assignment if teacher want to this in the university management system.

Scenarios

Post Conditions: None

29.4.6 Announcemid term

Description : This use case describes the teacher can announcemid-term this system

Actor : Teacher

Precondition : All actors must have the sign in identification
System shall allow to Teacher can announce the mid-term if teacher want to this in the university management system.

Scenarios

Post Conditions: None

29.4.7 Add Timetable

Description : This use case describes the teacher can add timetable this

	system
Actor :	Teacher
Precondition :	All actors must have the sign in identification
Scenarios	System shall allow to Teacher can add time table if teacher want to change the class or makeupclass in the university management system.
Post Conditions:	None

29.4.8 Calls Attendance

Description :	This use case describes the teacher can calls attendance the students this system
Actor :	Teacher
Precondition :	All actors must have the sign in identification
Scenarios	System shall allow to Teacher can calls the attendance of the students if teacher want to this in the university management system.
Post Conditions:	None

29.4.9 Add Results

Description :	This use case describes the teacher can add results the students this system
Actor :	Teacher
Precondition :	All actors must have the sign in identification
Scenarios	System shall allow to Teacher add results of the students if teacher want to this in the university management system.
Post Conditions:	None

29.4.10 Add notification

Description :	This use case describes the teacher can add notification for students this system
Actor :	Teacher

Precondition :	All actors must have the sign in identification System shall allow to Teacher can add notification for the students if teacher want to this in the university management system.
Scenarios	
Post Conditions:	None

29.5 Use case Of Student

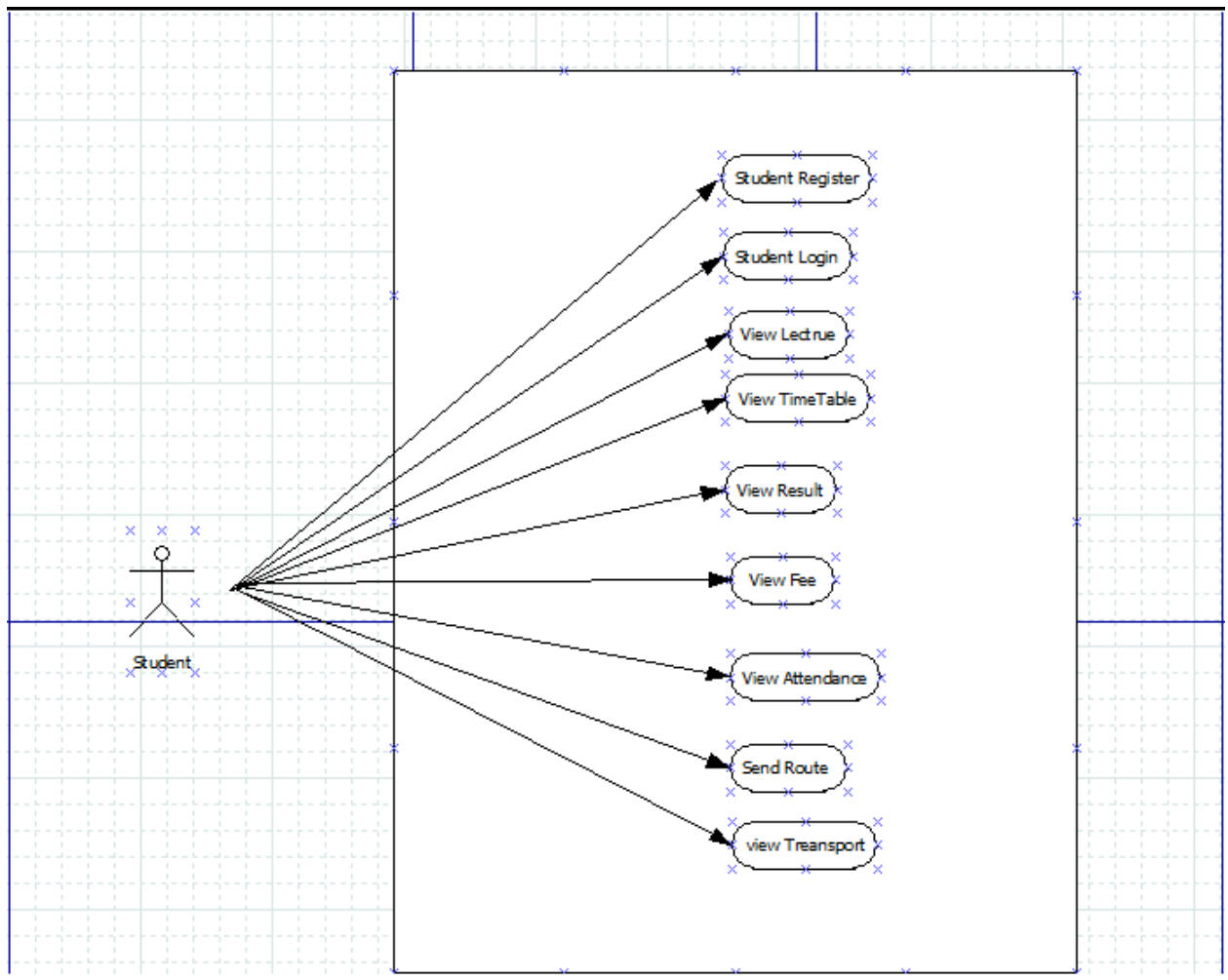


Figure 4

29.6 Explanation of use case of Students Panel

29.6.1 Login System

Description :	This use case describes the login system after login student can manage the system.
Actor :	student
Precondition :	All actors must have the sign in identification 1. The actor wants to use the application. 2. Click on ‘Sign In’ link on top of the homepage. a. If student doesn’t have an account, they must sign up.
Scenarios	3. Actor sign in to the application. 4. The authenticate system will validate the student id.
Post Conditions:	None

29.6.2 Student Registration

Description :	This use case describes the Admin can add New teacher in this system
Actor :	student
Precondition :	All actors must have the sign in identification System shall allow to student can register and view to their and student profile. Students can also view to their
Scenarios	own profile in online university management system.
Post Conditions:	None

29.6.3 Student view lectures

Description :	This use case describes the student can view lecture in this system
Actor :	student
Precondition :	All actors must have the sign in identification System shall allow to student can view lecture which teaches the teacher. Students can also view to their own
Scenarios	profile in online university management system.
Post Conditions:	None

29.6.4 Student view timetable

Description :	This use case describes the student can view timetable in this system
Actor :	student
Precondition :	All actors must have the sign in identification System shall allow to student can view timetable if teacher can assign time table for students. Students
Scenarios	can also view to their own profile in online university management system.
Post Conditions:	None

29.6.5 Student viewResults

Description :	This use case describes the student can view Results in this system
Actor :	student

Precondition :	All actors must have the sign in identification
Scenarios	System shall allow to student can view results if teacher can upload the results for students. Students can also view to their own profile in online university management system.

Post Conditions:	None
-------------------------	------

29.6.6 Student view fee

Description :	This use case describes the student can view fee in this system
----------------------	---

Actor :	student
----------------	---------

Precondition :	All actors must have the sign in identification
Scenarios	System shall allow to student can view fee if admin can upload the fee status for students. Students can also view to their own profile in online university management system.

Post Conditions:	None
-------------------------	------

29.6.7 Student view Attendance

Description :	This use case describes the student can view attendance in this system
----------------------	--

Actor :	student
----------------	---------

Precondition :	All actors must have the sign in identification
Scenarios	System shall allow to student can view attendance if teacher can upload the attendance for students. Students can also view to their own profile in online university management system.

Post Conditions:	None
-------------------------	------

29.6.8 Student send route

Description :	This use case describes the student can send route in this system
Actor :	student
Precondition :	All actors must have the sign in identification System shall allow to student can send route if admin accept the route for students. Students can also view to their own profile in online university management system.
Scenarios	
Post Conditions:	None

29.6.9 Student viewTransport

Description :	This use case describes the student can view transport in this system
Actor :	student
Precondition :	All actors must have the sign in identification System shall allow to student can view transports if admin active the route for students. Students can also view to their own profile in online university management system.
Scenarios	
Post Conditions:	None

30 Implementation methodology

Model view controller or MVC as it is popularly called is a software design pattern for developing web application. A model view controller is made up of following three parts

- Model- the lowest level of the pattern which is responsible for maintaining data
- View- this is responsible for displaying all or a portion of the data to the user
- Controller- software code that controls the interaction between the model and view

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the controller receives all request for the application and then works with middle to prepare any data needed by the view. The view then uses the data prepared by the controller to generate the final presentable response. The MVC abstraction can be graphically represented as follows.

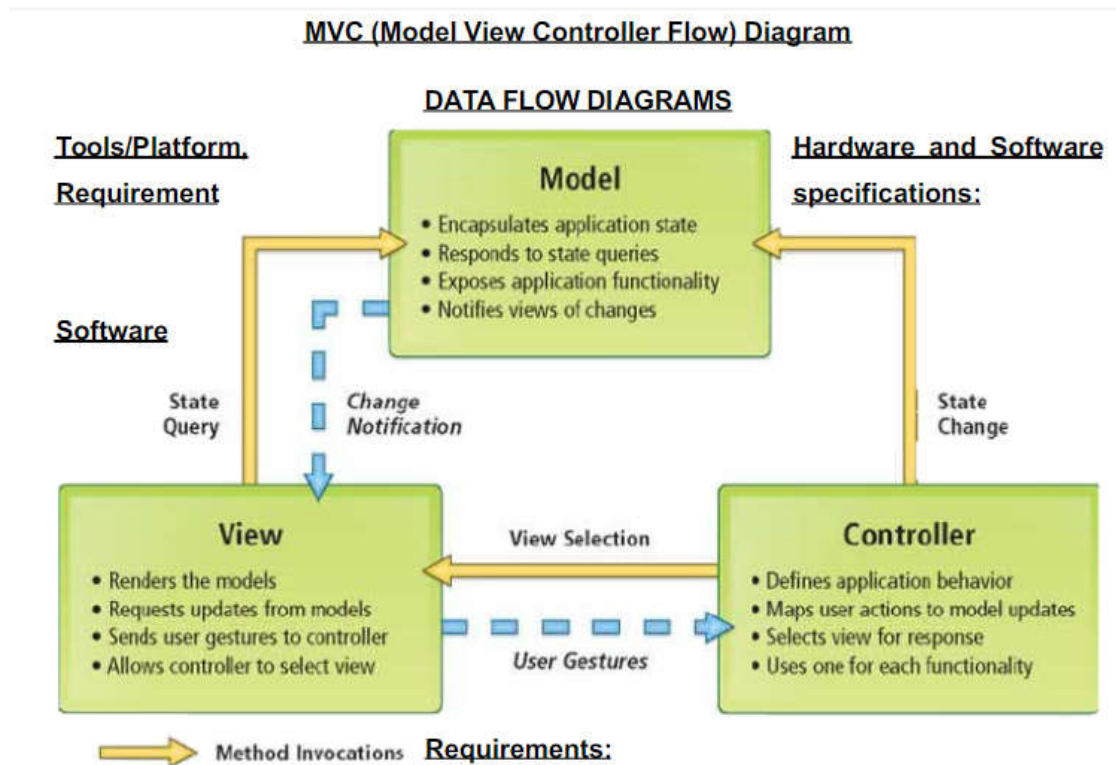


Figure 5

31 Data Flow Diagram

A data flow diagram[6] is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations.

A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

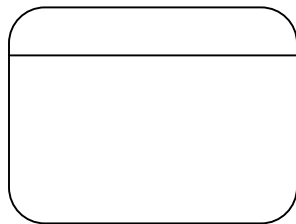
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this led to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

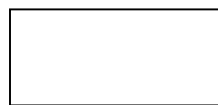
31.1 DFD Symbols

In the DFD, there are four symbols

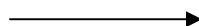
1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data



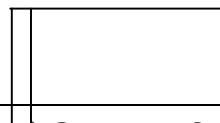
Process that transforms data flow.



Source or Destination of data



Data flow



Data Store

31.2 Constructing a DFD:

Several rules of thumb are used in drawing DFD'S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source.
3. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
4. When a process is exploded into lower level details, they are numbered.
5. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized.

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

31.3 Salient features of DFD'S

1. The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

31.4 Types of data flow diagrams

- 1.Current Physical
- 2.Current Logical
- 3.New Logical
- 4.New Physical

31.4.1 Current physical

In Current Physical DFD process label include the name of people or their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data. Similarly data flows and data stores are often labels with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

31.4.2 Current logical:

The physical aspects at the system are removed as much as possible so that the current system is reduced to its essence to the data and the processors that transforms them regardless of actual physical form.

31.4.3 New logical

This is exactly like a current logical model if the user were completely happy with the user were completely happy with the functionality of the current system but had problems with how it was implemented typically through the new logical model will differ from current logical model while having additional functions, absolute function removal and inefficient flows recognized.

31.4.4 New physical

The new physical represents only the physical implementation of the new system.

31.5 Rules governing the DFD'S

31.5.1 Process

- 1) No process can have only outputs.
- 2) No process can have only inputs. If an object has only inputs than it must be a sink.
- 3) A process has a verb phrase label.

31.5.2 Data store

- 1) Data cannot move directly from one data store to another data store, a process must move data.
- 2) Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into data store
- 3) A data store has a noun phrase label.

31.5.3 Source or sink

The origin and /or destination of data.

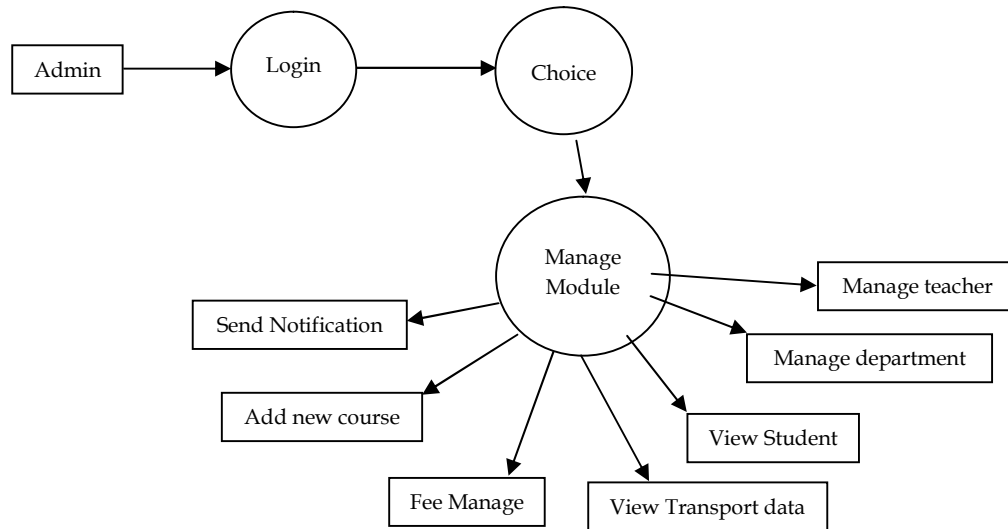
- 1) Data cannot move directly from a source to sink it must be moved by a process
- 2) A source and /or sink has a noun phrase label

31.5.4 Data flow

- 1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated however by two separate arrows since these happen at different times.
- 2) A join in DFD means that exactly the same data comes from any of two or more different processes, data store or sink to a common location.
- 3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow, produce some other data flow, returns the original data into the beginning process.
- 4) A Data flow to a data store means update (delete or change).
- 5) A data Flow from a data store means retrieve or use.

A data flow has a noun phrase label, more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

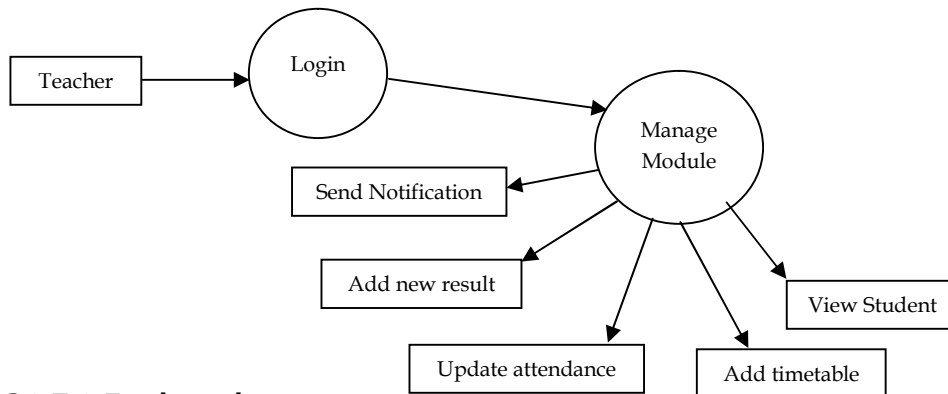
31.6 Admin



31.6.1 Explanation

Here admin can login with selection of his/her own role. After verification admin can manage his modules. Admin can add new programs to any of the department. It includes courses offered in the particular program in the Online University Management system. System shall allow to admin can view the student if student can register himself or herself in the university management system. Admin has all the rights which a teacher has with extra rights like add program, add teacher, add sessions, add courses, add courses etc. Meanwhile admin can remove, update and delete the above sentence.

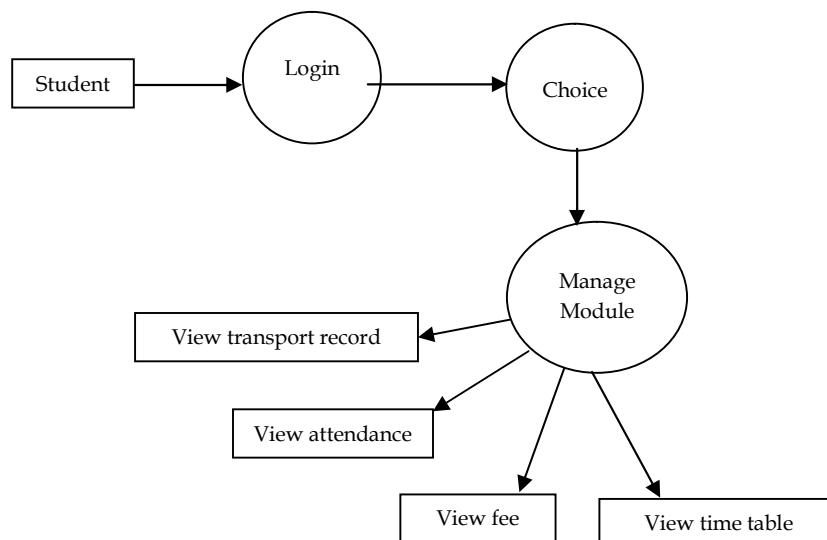
31.7 Teacher



31.7.1 Explanation

Here teacher can login with selection of his/her own role. After verification teacher can manage his modules. Teacher can update student record like papers, quizzes and assignment marks which can be viewed by related student.

31.8 Student



31.8.1 Explanation

Here student can login with selection of his/her own role. After verification student can view his modules. If student is not verified he has to register himself.

32ClassDiagram

Class diagram[7] is defining the object and information structures of users. It is defining the information without reference to any specific implementation. Its classes and relationships can be applied in database tables.

32.1 Admin

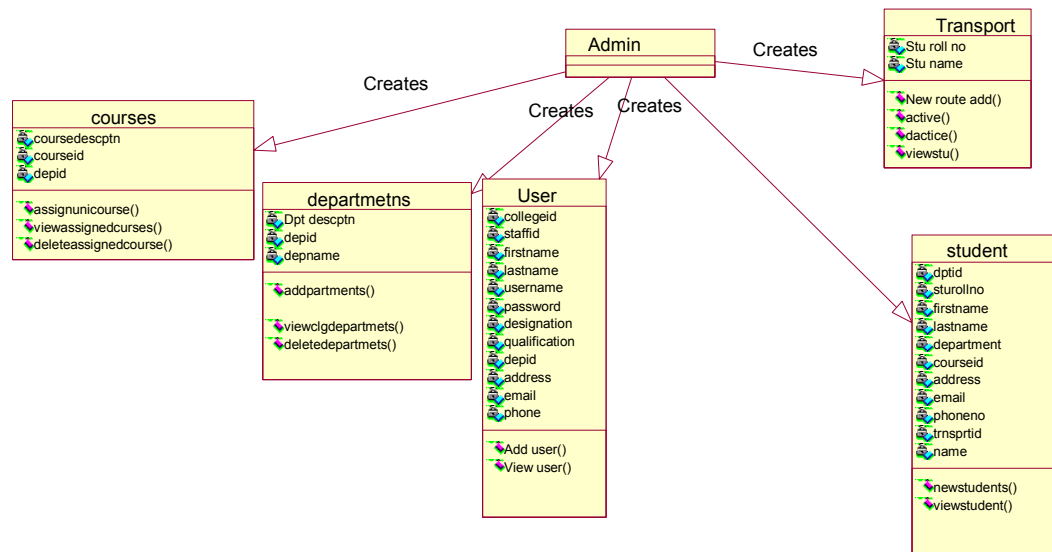


Figure 6-Class Diagram of admin

32.2 Student

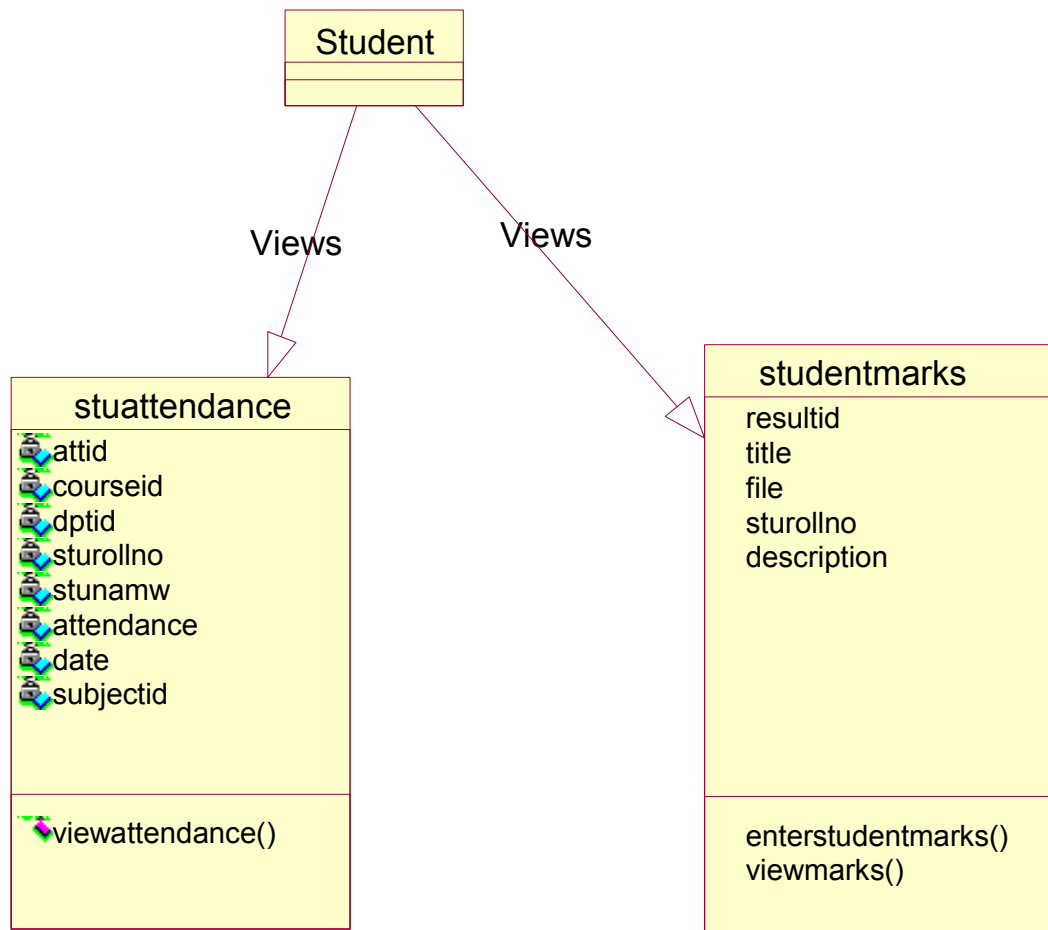


Figure 7-Class Diagram of student

33State Diagrams

State Diagrams[8]

33.1 Admin

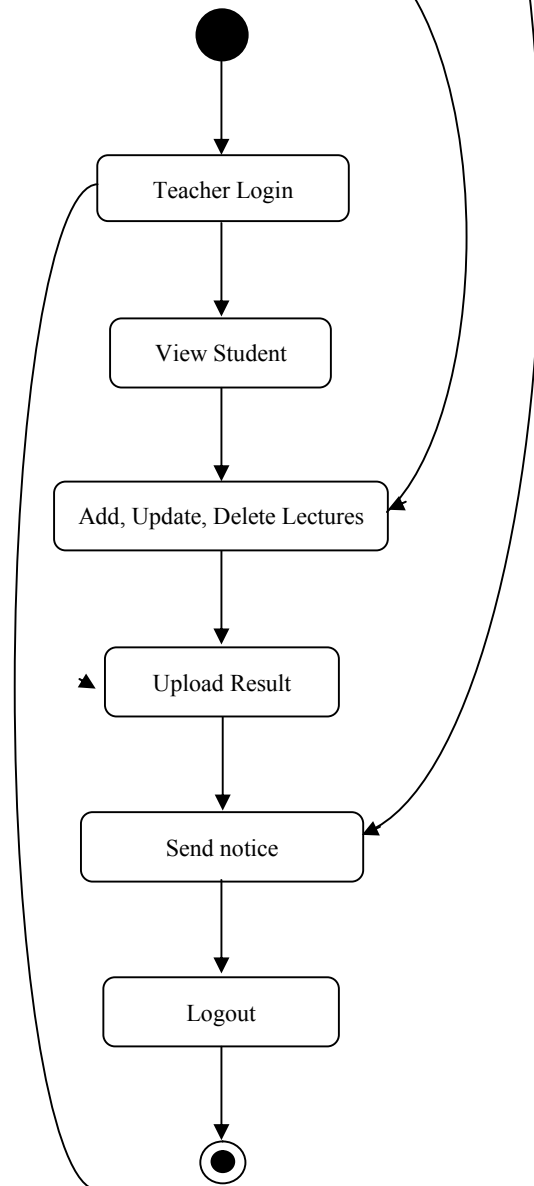


Figure 19.1

33.2 Student

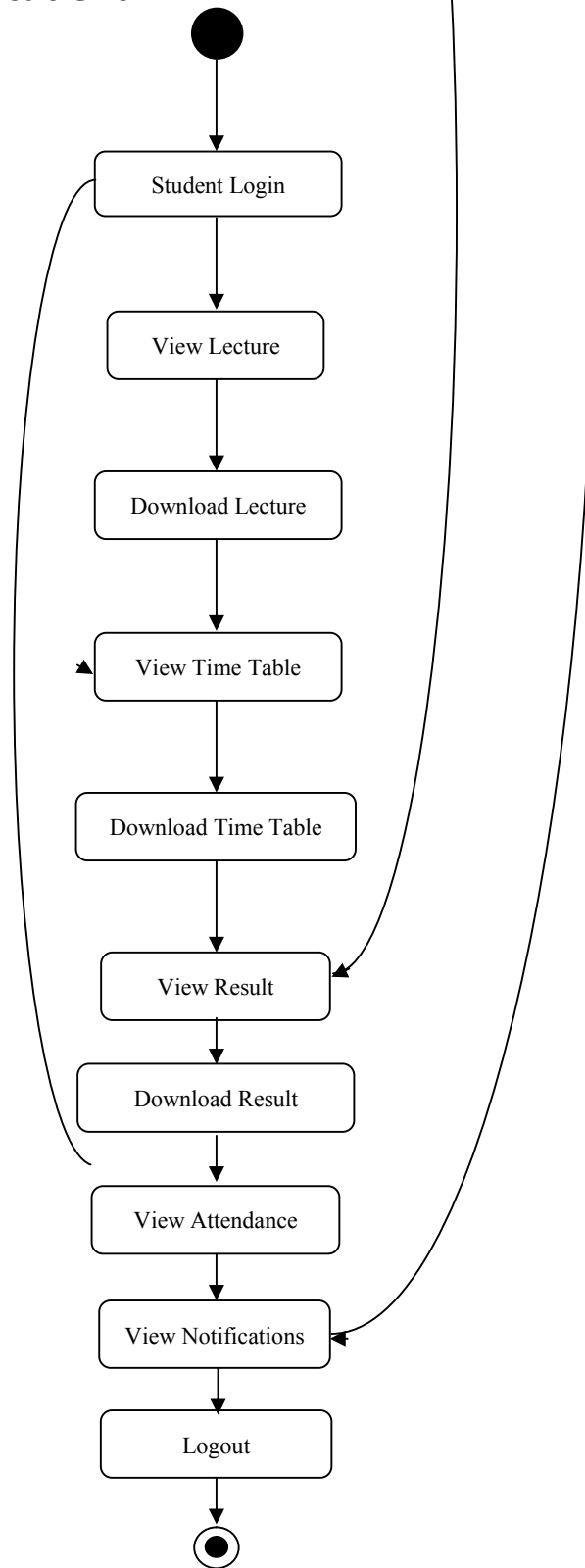


Figure 19.2

33.3 Teacher

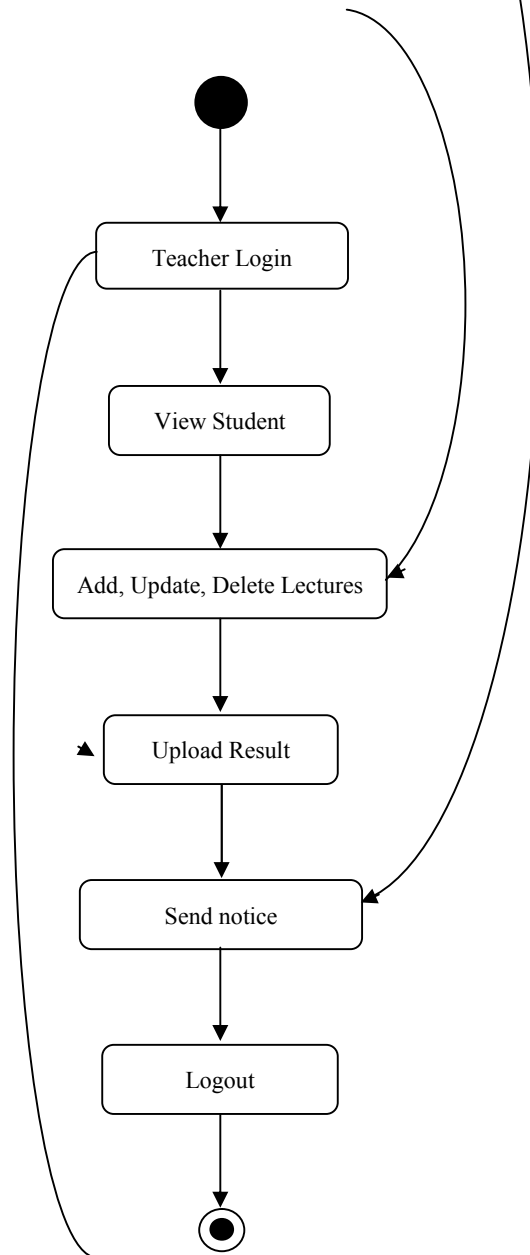


Figure 19.3

34 Database Tables

34.1 Users

34.1.1 Description

Here is list of users who has registered himself in university's website. Here is all login details which will be used to verification of user.

	Column Name	Data Type	Allow Nulls
	Id	int	☐
	First_Name	varchar(50)	☐
	Last_Name	varchar(50)	☐
	E_Mail	varchar(50)	☐
	Phone	varchar(50)	☐
	Password	varchar(50)	☐
	Confirm_Password	varchar(50)	☐
	Role	varchar(50)	☐
	Picture	varchar(MAX)	☐
	Status	varchar(50)	☐
	Rolnum	varchar(50)	☒
	Dept_nam	varchar(50)	☒
	Class	varchar(50)	☐
	InterMarks	varchar(50)	☒
	MatricMarks	varchar(50)	☒
	BloodGroup	varchar(50)	☒

Figure 8-database

	Column Name	Data Type	Allow Nulls
	id	int	☐
	Name	varchar(50)	☐

34.2 Departments

Figure 9-database

34.2.1 Description

In this section of the database Admin can enter new departments in the database.

Campuses may be in the same city or may be in the different cities with having particular university Id.

34.3 Course

	Column Name	Data Type	Allow Nulls
	Id	int	☐
	Title	varchar(50)	☐
	Course	varchar(50)	☐
	Description	varchar(MAX)	☐

Figure 10-database

34.3.1 Description

In this section of the database, the user that is admin can add new course to any of the department. It includes courses offered in the particular program in the Online University Management system

Programs are necessary for each of the campus of the said university.

34.4 Student Info

Figure 11-database

	Column Name	Data Type	Allow Nulls
	Id	int	☐
	First_Name	varchar(50)	☐
	Last_Name	varchar(50)	☐
	E_Mail	varchar(50)	☐
	Phone	varchar(50)	☐
	Password	varchar(50)	☐
	Confirm_Password	varchar(50)	☐
	Role	varchar(50)	☐
	Picture	varchar(MAX)	☐
	Status	varchar(50)	☐
	Rolnum	varchar(50)	☒
	Dept_nam	varchar(50)	☒
	Class	varchar(50)	☐
	InterMarks	varchar(50)	☒
	MatricMarks	varchar(50)	☒
	BloodGroup	varchar(50)	☒

34.4.1 Description

This section of the database of University Management system the database saves the personal record of the student who is willing to get admission in the university.

This record of the student will be referred to the department and will always remain in the record of the said university.

34.5 Student Courses

	Column Name	Data Type	Allow Nulls
	Id	int	☐
	Title	varchar(50)	☐
	Course	varchar(50)	☐
	Description	varchar(MAX)	☐

Figure 12-database

34.5.1 Description

This section of the database of University Management system the database saves the record of the courses in which a student is enrolled by the university.

This record contains the student unique id that is assigned by the university in which the student is now with including the courses he has opted for the said university.

	Column Name	Data Type	Allow Nulls
	Id	int	☐
	Title	varchar(500)	☐
	[File]	varchar(500)	☐
	Description	varchar(MAX)	☐

34.6Results

Figure 13-database

34.6.1 Description

After having completion of the course or even of the degree, this section of the database of University Management system the database saves the record of the result of the student.

This shows the complete record of any of the result including his detailed marks sheet.

34.7 Fee Info

	Column Name	Data Type	Allow Nulls
	Id	int	☐
	Student_Name	varchar(50)	☐
	RollNum	varchar(50)	☐
	Subject	varchar(50)	☐
	Fee	varchar(50)	☐
	Date	varchar(50)	☐

Figure 9-database

34.7.1 Description

This section of the database of University Management system the database saves the record of the fee of the student.

This may include the admission fee and also the record of the missing fee. Student may be granted concession in the fee.

The wholesome fee is saved in this table and the status of the student is declared.

35 Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The

underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems

35.1 Strategic approach to software testing

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

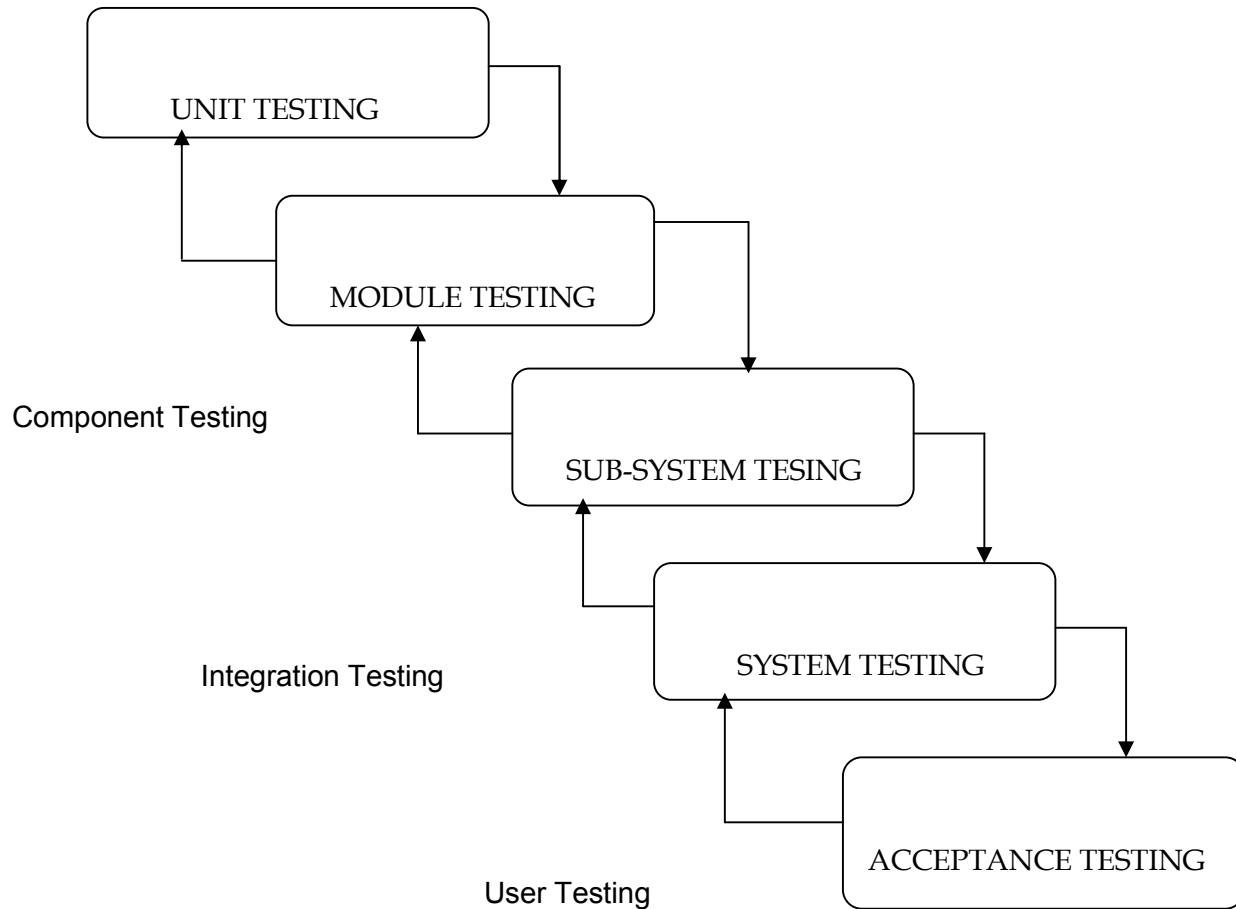


Figure 14- Strategic approach to software testing

35.2 Unit testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

35.3 White box testing

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

35.4 Basic path testing

Established technique of flow graph with Cyclamate complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

- Use the design of the code and draw correspondent flow graph.
- Determine the Cyclamate complexity of resultant flow graph, using formula:

$$V(G)=E-N+2 \text{ or}$$

$$V(G)=P+1 \text{ or}$$

$$V(G)=\text{Number Of Regions}$$

Where $V(G)$ is Cyclamate complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

35.6 Conditional testing

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

35.7 Data flow testing

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

35.8 Loop testing

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of connected loop.
- Unstructured loops were resolved into nested loops or concatenated loops and tested as above.
- Each unit has been separately tested by the development team itself and all the input have been validated.

36 User Interfaces

36.1 User Interface Design:

It is essential to consult the system users and discuss their needs while designing the user interface:

36.2 User Interface Systems can be broadly classified as:

1. User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
2. Computer initiated interfaces

In the computer initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

36.3Login

Welcome, Please login

E-Mail

Password

☐ Admin ☐ Teacher ☐ Student

Login

© 2017 Student Signup

36.3.1 Description:

Admin, teacher or student can login by using their role and typing correct Email and password to access the system.

36.4Admin

36.4.1Add courses

New Course

Name

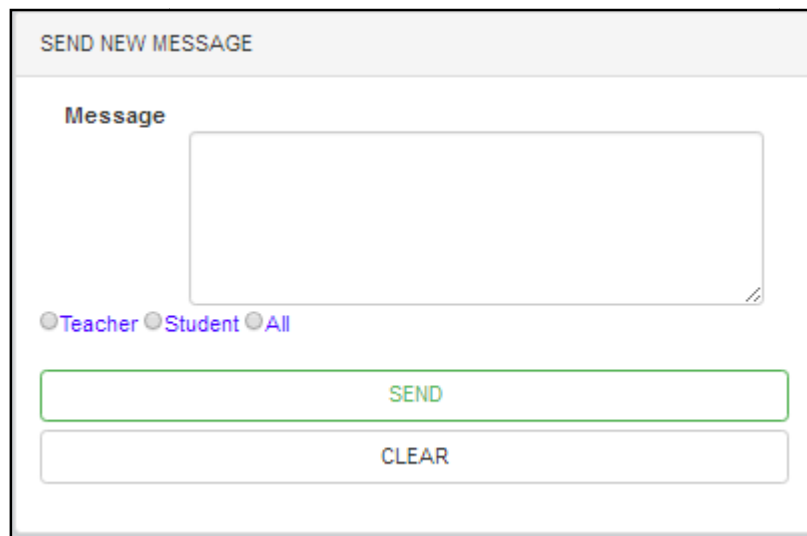
Save Reset

Figure 16-User Interface

36.4.1.1 Description

System shall allow to admin add courses in online university management system.

36.4.2SendNotification



The interface is titled "SEND NEW MESSAGE". It features a text area labeled "Message" for composing the message. Below the text area are three radio buttons for selecting the recipient: "Teacher", "Student", and "All". At the bottom, there are two buttons: "SEND" and "CLEAR".

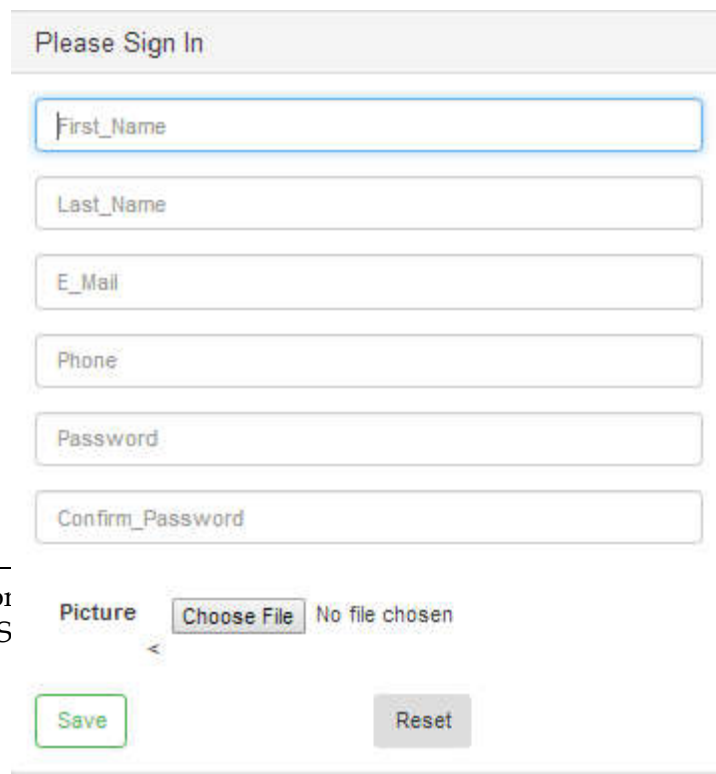
Figure 17-User Interface

Admin write the new message that he want to send. He will be send to management or teachers. If he doesn't want to send, he can clear the message.

36.4.2.1 Description

System shall allow to admin to send notification on online university management system.

36.4.3Add



The form is titled "Please Sign In". It contains several input fields: "First_Name", "Last_Name", "E_Mail", "Phone", "Password", and "Confirm_Password". Below these fields is a "Picture" section with a "Choose File" button and the text "No file chosen". At the bottom, there are "Save" and "Reset" buttons.

New Admin

Figure 18-User Interface

36.4.3.1 Explanation

Only Admin has authority to add new admin. When Admin will make new admin, it require following information:

- Name
- E-mail
- Phone number etc.

Admin enter the information like (First name, last name, email, phone number, password and pictures etc.) after this save the information.

If he doesn't want to add new admin then clear the information and reset it.

System shall allow to admin add new admin in online university management system.

36.4.4 Dashboard

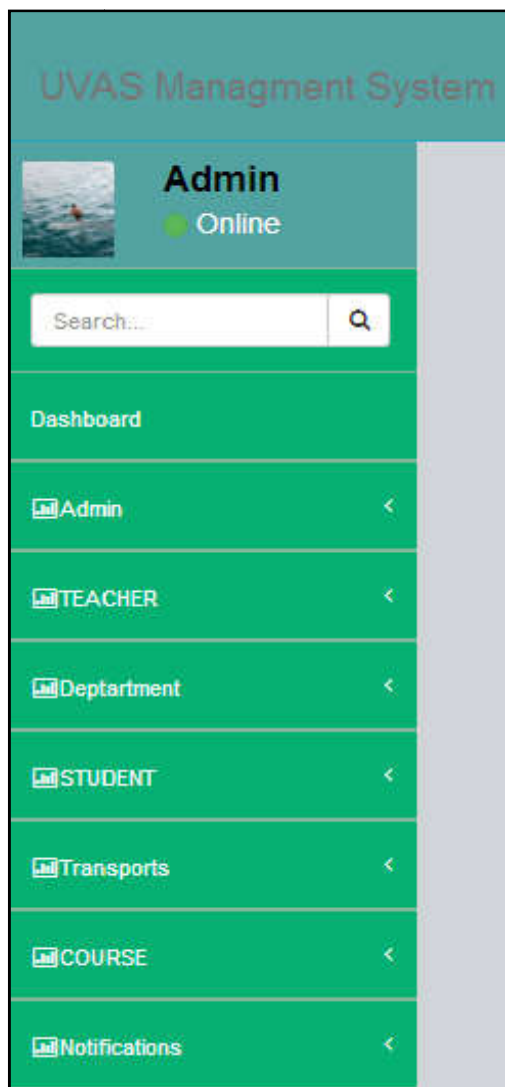


Figure 19-User Interface

It has a dashboard and search bar. In dashboard just Admin can view and get the information about following:-

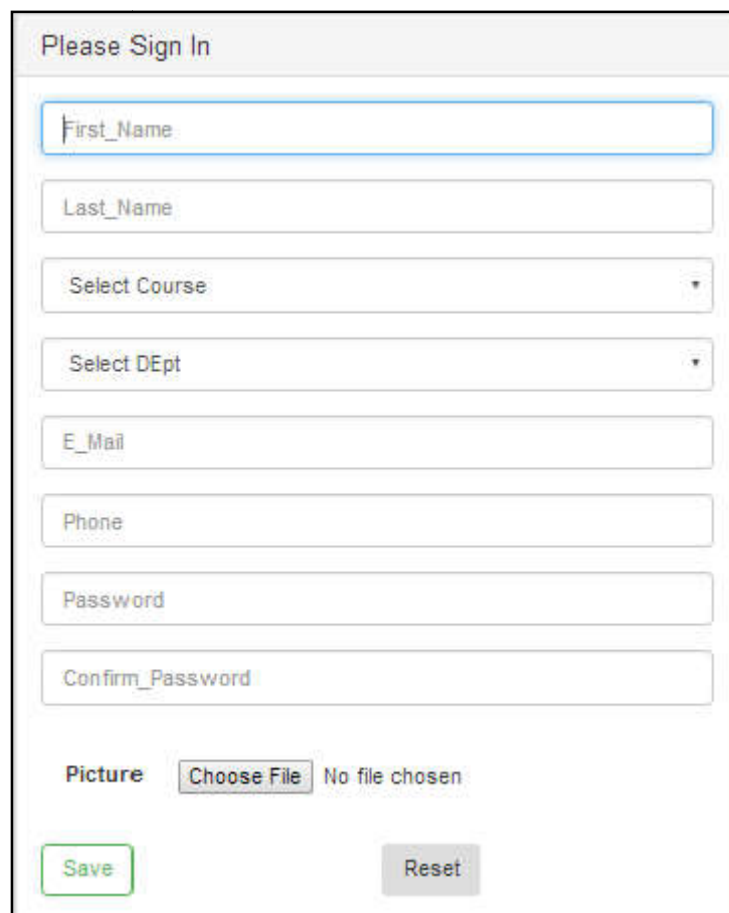
- Admin
- Teachers
- Department
- Students
- Course
- Transport
- Notification.

Admin can add new Teacher give any news any update etc. about university. Admin has authority to anything about university, teachers, students etc.

He can get information about any teacher, He can add new teacher, and he can check teacher's performance.

- Admin can also check performance of any department.
- He can also view the total strength of students and check their performance.
- Admin can also add new course of any department.
- He can check how many students use university transport.
- He can send notification about any update etc. to management.

36.4.5 Add new Teacher



Please Sign In

First_Name

Last_Name

Select Course

Select DEpt

E_Mail

Phone

Password

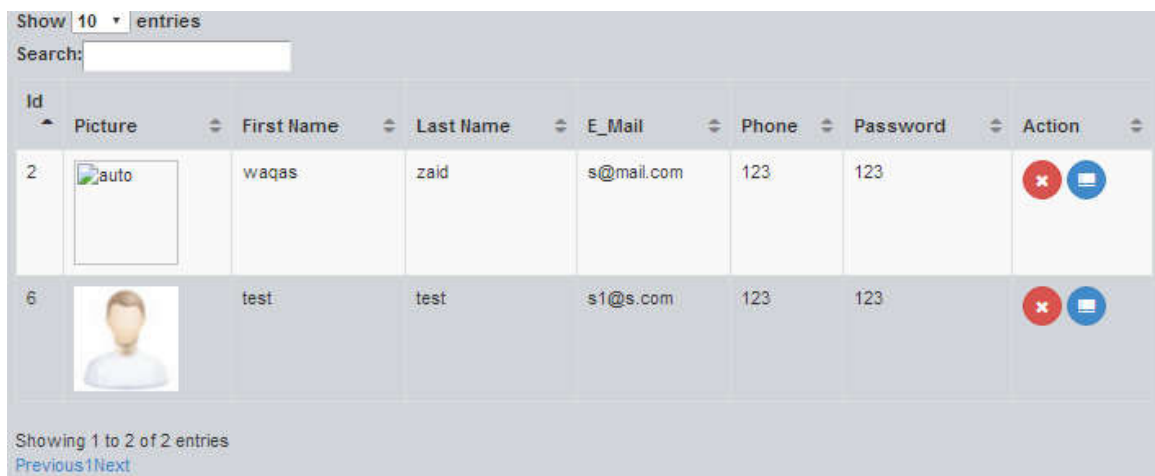
Confirm_Password

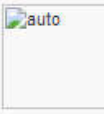
Picture No file chosen

36.4.5.1 Description

System shall allow to admin add new teacher in online university management system.

36.4.6 View Students



Id	Picture	First Name	Last Name	E-Mail	Phone	Password	Action
2		waqas	zaid	s@mail.com	123	123	 
6		test	test	s1@s.com	123	123	 

Showing 1 to 2 of 2 entries
[Previous](#) [Next](#)

Figure 21-User Interface

36.4.6.1 Description

System shall allow to admin to view all registered students in online university management system.

36.5 Teacher

Teachers can view this dashboard. It consists of following:

- Students
- Lectures
- Timetable
- Attendance
- Results

- Notification

Teachers can view the total strength of the class. They can give assignments and quiz etc. to students.

They can provide teaching materials to students.

Students can get online, if notes are missing.

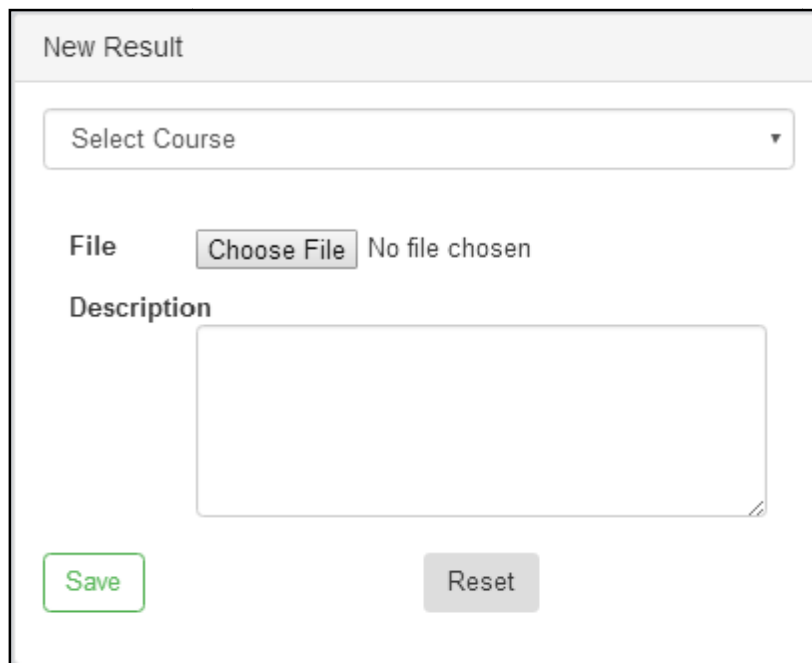
Teachers can make timetable on system instead of manual.

The attendance will be manage online instead of register.

After preparing the results, Teachers can announced online instead of display on board.

Teachers will be able to send notification about anything like (quiz, assignment etc.) to students.

36.5.1 Add Result



The screenshot shows a web form titled "New Result". It contains a dropdown menu labeled "Select Course". Below this is a "File" section with a "Choose File" button and the text "No file chosen". Underneath is a "Description" label followed by a large text area. At the bottom left is a green "Save" button, and at the bottom right is a grey "Reset" button.

Figure 22-User Interface

36.5.1.1 Description

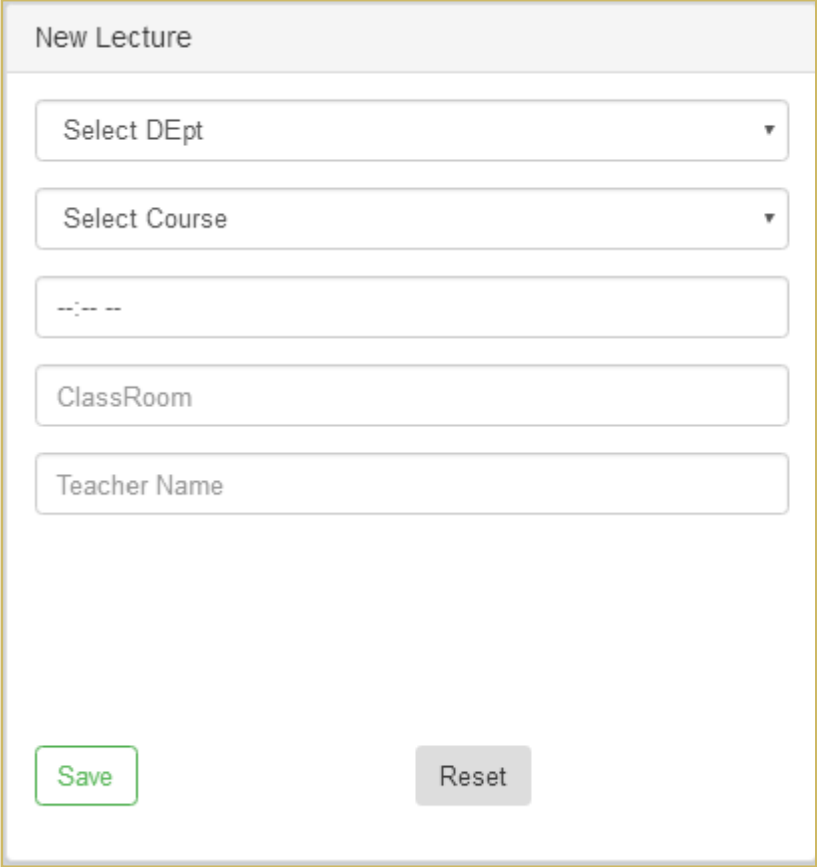
System shall allow teacher to add results of students in online university management system.

36.5.2 Send Notification

The image shows a user interface for sending a new message. It consists of a light gray header bar with the text "SEND NEW MESSAGE". Below the header, the word "Message" is displayed in bold. To the right of "Message" is a large, empty rectangular text input field. Below the input field are two buttons: a green button with the text "SEND" and a white button with the text "CLEAR".

Figure 23-User Interface

36.5.3 Add Time Table

A screenshot of a web form titled "New Lecture". The form is enclosed in a light gray border. It contains five input fields stacked vertically: a dropdown menu labeled "Select DEpt", another dropdown menu labeled "Select Course", a date input field with a placeholder "--:-- --", a text input field labeled "ClassRoom", and a text input field labeled "Teacher Name". At the bottom left of the form is a green "Save" button, and at the bottom right is a gray "Reset" button.

New Lecture

Select DEpt ▼

Select Course ▼

--:-- --

ClassRoom

Teacher Name

Save Reset

Figure 24-User Interface

36.5.3.1 Explanation

Admin can add new timetable.in which he can add following:

- New lecture for selected teachers
- Department
- Class Room
- Course
- Teacher Name

He can add this information. He can also reset this information.It consist of search bar.

User (students or teachers etc.) can search any course that they want get information like timing, teacher name, subject name etc.

36.5.4 Add Attendance

Select Department

BSCS

BBA

Select Class

OPP

MATH

physics

Students

Student	RollNum	Class	DPT	Present	Absent	Leave
---------	---------	-------	-----	---------	--------	-------

SUBMIT

Figure 25-User Interface

In this option, teacher can select following:

- Department
- Class
- Subject
- Students

Then teacher start the students.

Student option following:

- Roll

After calling teacher will mark it or send to

36.5.5Dashboard

the attendance of

further divided into

- Name
- Number
- Department
- Present
- Absent
- Leave

students attendance, and finally submit management.

Figure 26-User Interface

36.5.6 View Notification

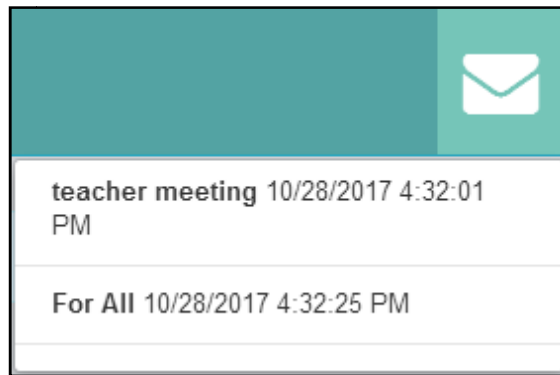


Figure 27-User Interface

When the admin send notification related to anything like meeting etc. to teachers. They (Teachers) can view it and they'll attend the meeting.

36.5.7 Time able

36.6Student

Show 10 entries

Search:

Id	Course	Teacher	Class	Time	Action
1	OPP	tets	bb2	02:00:00	

Showing 1 to 1 of 1 entries

[Previous](#)
[Next](#)

Figure 28-User Interface

When the students online.It view the dashboard that consists of following:

- Lecture
- Fee
- Timetable,
- Result,
- Attendance
- Transport

It help the students to get help without face any problem.If the students absent or if they can't attend class due to any reason this system provide help to get materiel.

- If they does not have timetable they can get online or if any update about timetable from Management, students received notification.
- When the result will announce, students receive notification and they can view their result online without going university.
- Attendance will be online.
- Transport will be also manage by system.
- Students can get information about fare and timing of convince stop.

36.6.1View/Download Results

Show 10 entries

Search:

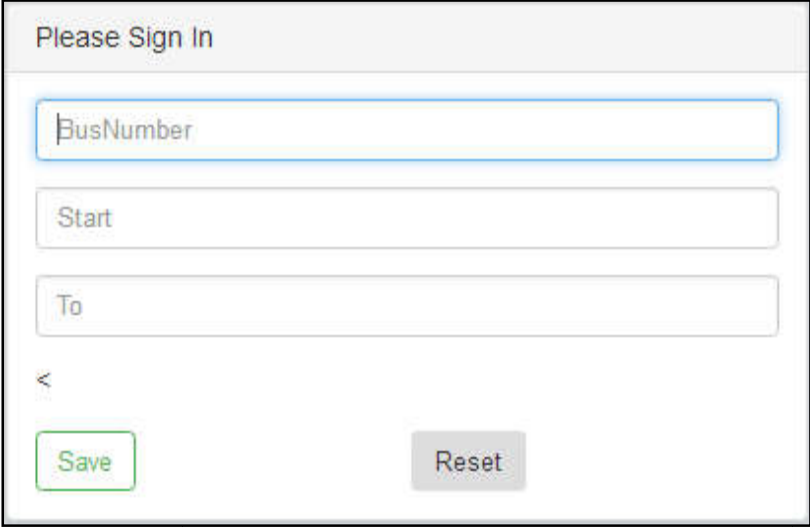
Id	Course	Teacher	Class	Time
1	OPP	tets	bb2	02:00:00
2	OPP	aslam	45	02:25:00
3	MATH	aa	4	05:41:00

Showing 1 to 3 of 3 entries

[Previous](#)
[Next](#)

Figure 30-User Interface

36.6.3Transport



The image shows a web-based user interface for a transport system. It features a light gray header with the text "Please Sign In". Below the header, there are three input fields: the first is labeled "BusNumber" and has a blue border; the second is labeled "Start" and the third is labeled "To". Below these fields is a small left-pointing arrow icon. At the bottom of the form, there are two buttons: a green "Save" button on the left and a gray "Reset" button on the right.

Figure 31-User Interface

Student can view the transport system and can do any changes. If they want to get information regarding bus. Like (bus timing), he can save information and he also has authority to reset.

37 System Security

The protection of computer-based resources that includes hardware, software, data, procedures and people against unauthorized use or natural disaster is known as **System Security**.

System Security can be divided into four related issues:

- Security
- Integrity
- Privacy
- Confidentiality

System security refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

Data security is the protection of data from loss, disclosure, modification and destruction.

System integrity refers to the proper functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

Privacy defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

Confidentiality is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

37.1 Security in software

System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employees two types of checks and controls:

37.1.1 Client side validation

Various client side validations are used to ensure on the client side that only valid data is entered. Client side validation saves server time and load to handle invalid data. Some checks imposed are:

- VBScript is used to ensure those required fields are filled with suitable data only. Maximum lengths of the fields of the forms are appropriately defined.
- Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.

- Tab-indexes are set according to the need and taking into account the ease of user while working with the system.

37.1.2 Server side validation

Some checks cannot be applied at client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed is:

- Server side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
- User is intimating through appropriate messages about the successful operations or exceptions occurring at server side.
- Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the organizational structure.
- Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled o the server side.
- Using server side validation, constraints on several restricted operations are imposed.

38 Maintenance

Maintenance the last phase in the software engineering process. As programs are developed.

A distributing trend has emerged the amount of effort and a resource expended on software maintenance is growing. In total project development maintenancetakes 65% of effort .In software maintenance there are four. They are:

- Adaptive Maintenance
- Corrective Maintenance

- Perfective Maintenance
- Preventive Maintenance

Adaptive Maintenance is applied when changes in the external environment precipitate modifications to software. It deals with adapting the software to new environments.

Perfective Maintenance incorporates enhancements that are requested by user community. It deals with updating the software according to changes in user requirements.

Corrective Maintenance acts to correct errors that are uncovered after the software is in use. It deals with fixing bugs in the code

Preventive Maintenance improves future maintainability and reliability and provides a basis for future enhancement. It deals with updating documentation and making the software more maintainable. Tasks performed during the software engineering process define maintainability and have an important impact in the success of any maintenance approach. Reverse Engineering and Reengineering are the tools and techniques used to maintain the project.

There are four major problems that can slow down the maintenance process

- Unstructured code
- Maintenance programmers having insufficient knowledge of the System
- Documentation being absent
- Out of Date , or at best insufficient

The success of the maintenance phase relies on these problems being earlier in the life cycle.

39 Conclusion

The online university management system allows the admin to manage the record of the student. It facilitates the students to sign up from their home. It would enable the educational institutes to register the students online from any online machine. It is much securing web application because all the programming is on the database end [9].

After creating login by student on university management system, if the faculty accepts his/her request otherwise would not be appeared on the university record. Further the Teacher has a right to register a student after getting login. Admin has all the rights which a teacher has with extra rights like add program, add teacher, add sessions, add courses, add courses etc. Meanwhile admin can remove, update and delete the above sentence.

After having registered, a student can see his/her profile on being logged in. He can see his results and fee status. He can choose his courses. It is true that further amendments could be made in the future era for more perfection but this program is now capable of deployment as on having this testing. This system has developed for online University Management System to keep all required information safe and which can be maintained easily. This program can store all the record as well as results of all the students.

40Limitation

Deactivated users or unregistered user cannot login. They must belong to activated or registered user. The result must be displayed in their login panel, student can only see their own result and cannot access the other student result. And can only see his own result and profile. This could be easily implemented while programming the separate panel. Security logs though not implemented on this system would be well available through the respective database management system and web server software.

41Future Work

Future work the system can be enhanced to support other types of model like online examination test[10], accounts etc. the system can be improved by deploying it to the World Wide Web, so user can use system from different places. The current system can support user login at a time from the server. The current system works only a limited university functionality, in future the system works for all departments, each department will have only one admin like as program manager[11]. That admin handled the all task

of own department. Current system handles small unit of users, in future system holds large number of users of each department like class advisor [12], department coordinators [13] etc.

42References

[[Online]. Available: <http://searchsqlserver.techtarget.com/definition/SQL-Server>.
1 [Accessed 06 11 2017].
]

[[Online]. Available: <http://cybervision.com.pk/university-management-system.php>.
2 [Accessed 06 11 2017].
]

["Learn MVC," [Online]. Available:
3 <https://www.codeproject.com/Articles/866143/Learn-MVC-Project-in-days-Day>.
] [Accessed 06 11 2017].

[[Online]. Available:
4 https://www.researchgate.net/publication/236846274_Web_portal_As_A_Knowledg
] e_Management_System_In_The_Uni_versities. [Accessed 06 11 2017].

["Use Case Diagram," [Online]. Available: <https://msdn.microsoft.com/en-us/library/dd409432.aspx>. [Accessed 06 11 2017].
]

["DFD," [Online]. Available: <https://www.lucidchart.com/pages/data-flow-diagram/how-to-make-a-dfd>. [Accessed 06 11 2017].
]

["Class Diagram," [Online]. Available: <https://www.smartdraw.com/class-diagram/>.
7 [Accessed 06 11 2017].
]

[[Online]. Available: <https://www.lucidchart.com/pages/uml/activity-diagram>.
8 [Accessed 06 11 2017].
]

[W. Arif and S. Ahmad, "www.knowledge.com," 3 Nov 2017. [Online]. Available:
9 www.uvas.edu.pk.
]

[[Online]. Available: <https://www.onlineexambuilder.com/>. [Accessed 06 11 2017].

1
0
]

[[Online]. Available:

1 https://www.google.com.pk/search?dcr=0&source=hp&ei=ODkAWsn8JtDVwAKZ07iIDw&q=program+manager+job+description&oq=program+mana&gs_l=psy-ab.3.1.0l10.2067.7856.0.9979.12.12.0.0.0.0.247.1347.2-6.6.0....0...1.1.64.psy-ab..6.6.1346...0i131k1.0.eHa2eiBJPzQ. [Accessed 06 11 2017].

[[Online]. Available:

1 https://www.google.com.pk/search?dcr=0&ei=QzkAWp2MGIKHajS3nvAG&q=class+adviser+job+description&oq=class+adv+job+description&gs_l=psy-ab.3.0.0i7i30k1l2j0i7i5i30k1.22837.24619.0.25638.9.9.0.0.0.0.250.910.2-4.4.0....0...1.1.64.psy-ab..5.4.909....0.GjvdHlJ7Vb. [Accessed 06 11 2017].

[[Online]. Available:

1 https://www.google.com.pk/search?dcr=0&ei=XjkAWrDmDoG1a4HlkNAG&q=coordinator+job+description&oq=coordinat&gs_l=psy-ab.3.1.0i67k1l7j0l3.15195.17082.0.19240.9.9.0.0.0.0.397.843.2-2j1.3.0....0...1.1.64.psy-ab..6.3.842....0.8mWzKrAey-I. [Accessed 06 11 2017].

["Table of content," 06 11 2017. [Online]. Available:

1 https://www.google.com.pk/search?dcr=0&source=hp&ei=WDUAWomiJoHTwQL51KOwAQ&q=how+to+make+page+numbers+start+on+page+3&oq=how+to+make+page+numbers&gs_l=psy-ab.3.3.0l10.47417.111073.0.115900.24.22.0.0.0.0.566.2929.2-4j3j0j2.9.0....0...1.1.64.psy-ab..15.9.29.

["Learn SQL," [Online]. Available:

1 https://www.google.com.pk/search?dcr=0&source=hp&ei=WDUAWomiJoHTwQL51KOwAQ&q=how+to+make+page+numbers+start+on+page+3&oq=how+to+make+page+numbers&gs_l=psy-ab.3.3.0l10.47417.111073.0.115900.24.22.0.0.0.0.566.2929.2-4j3j0j2.9.0....0...1.1.64.psy-ab..15.9.29. [Accessed 06 11 2017].