**Project Report**

**On**

**“University Management System”**

Submitted in the Partial fulfillment of the requirement for the Award of Degree of

**Bachelor of Technology**

in

**COMPUTER SCIENCE & ENGINEERING**

Batch

(2017-2021)



**Subject-Software Engineering**

**(ACCS-16504)**

|  |  |
| --- | --- |
| **Submitted to**  **Er. Bhuvnesh Kumar** | **Submitted by**  Aayush Khanna (1701214)  Gopal Raichand (1701259)  Gulmehak Kaur (1701260) |

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**AMRITSAR GROUP OF COLLEGES, Amritsar**

**(Autonomous college under UGC Act – 1956[2(f) and 12(B)]**

**ACKNOWLEDGEMENT**

This is a humble effort to express our sincere gratitude towards those who have guided and helped us to complete this project.

A project is major milestone during the study period of a student. As such this project was a challenge to us and was an opportunity to prove our caliber. We are highly grateful and obliged to each and everyone making us help out of problems being faced by us.

It would not have been possible to see through the undertaken project without the guidance of

Er. Bhuvensh Kumar. It was purely on the basis of their experience and knowledge that we able to clear all the theoretical and technical hurdles during the development phases of this project work.

Last but not the least we are very thankful to our Head of Department Er. Vinod Sharma and all Members of Computer Science Department, who gave us an opportunity to face real time problems while fulfilling need of an organization by making projects for them.

**DECLARATION**

We hereby declare that the project work entitled **“University Management System”** is an authentic record of our work carried out as requirements of Institutional Training project for the award of degree of B.Tech(CSE), **Amritsar Group Of Colleges, Amritsar,** under the guidance of Er. Bhuvensh Kumar

(Signature of student)

**Aayush Khanna (1701214)**

**Gopal Raichand (1701259)**

**Gulmehak Kaur(1701260)**

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

**Faculty Coordinator**

Er. Bhuvnesh Kumar (Assistant Professor – CSE Department)

INDEX PAGE

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Content** | Page No. |
| 1 | Introduction to the Subject | 6 |
| 2 | Introduction to Project | 15 |
| 3 | Software Requirements Specification (SRS) | 17 |
| 4 | Data Flow Diagrams (DFD) | 26 |
| 5 | Unified Modeling Language (UML) Diagrams | 28 |
| 6 | Unit Testing and Integration Testing | 32 |

**INTRODUCTION TO SUBJECT**

# Software Engineering

Software engineering is an engineering discipline that’s applied to the development of software in a systematic approach (called a software process).

It’s the application of theories, methods, and tools to design build a software that meets the specifications efficiently, cost-effectively, and ensuring quality.

It’s not only concerned with the technical process of building a software, it also includes activities to manage the project, develop tools, methods and theories that support the software production.

Not applying software engineering methods results in more expensive, less reliable software, and it can be vital on the long term, as the changes come in, the costs will dramatically increase.

Different methods and techniques of software engineering are appropriate for different types of systems. For example, games should be developed using series of prototypes, while critical control systems require a complete analyzable specification to be developed.

## Computer Science Vs Software Engineering

Computer science focuses on the theory and fundamentals, like algorithms, programming languages, theories of computing, artificial intelligence, and hardware design, while software engineering is concerned with the activities of developing and managing a software.

# Software Engineers

The job of a software engineer is difficult. It has to balance between different people involved, such as:

* **Dealing with users**: User don’t know what to expect exactly from the software. The concern is always about the ease of use and response time.
* **Dealing with technical people**: Developers are more technically inclined people so they think more of database terms, functionality, etc.
* **Dealing with management**: They are concerned with return on their investment, and meeting the schedule.

**Dual Role of Software:**

**1. As a product –**

* It delivers the computing potential across network of Hardware.
* It enables the Hardware to deliver the expected functionality.
* It acts as information transformer because it produces, manages, acquires, modifies, displays, or transmits information.

**2. As a vehicle for delivering a product –**

* It provides system functionality (e.g., payroll system)
* It controls other software (e.g., an operating system)
* It helps build other software (e.g., software tools)

**Objectives of Software Engineering:**

1. **Maintainability**  
   It should be feasible for the software to evolve to meet changing requirements.
2. **Correctness**  
   A software product is correct, if the different requirements as specified in the SRS document have been correctly implemented.
3. **Reusability**

A software product has good reusability, if the different modules of the product can easily be reused to develop new products.

1. **Testability**

Here software facilitates both the establishment of test criteria and the evaluation of the software with respect to those criteria.

1. **Reliability**  
   It is an attribute of software quality. The extent to which a program can be expected to perform its desired function, over an arbitrary time period.
2. **Portability**  
   In this case, software can be transferred from one computer system or environment to another.
3. **Adaptability**  
   In this case, software allows differing system constraints and user needs to be satisfied by making changes to the software.

**Program vs Software Product:**

1. Program is a set of instruction related each other where as Software Product is a collection of program designed for specific task.
2. Programs are usually small in size where as Software Products are usually large in size.
3. Programs are developed by individuals that means single user where as Software Product are developed by large no of users.
4. In program, there is no documentation or lack in proper documentation.  
   In Software Product, Proper documentation and well documented and user manual prepared.
5. Development of program is Unplanned, not Systematic etc but Development of Software Product is well Systematic, organised, planned approach.
6. Programs provide Limited functionality and less features where as Software Products provides more functionality as they are big in size (lines of codes) more options and features.

# SOFTWARE TESTING

### Introduction:-

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free.If testing is done successfully it will remove all the errors from the software.

### Principles of Testing:-

(i) All the test should meet the customer requirements  
(ii) To make our software testing should be performed by third party  
(iii) Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.  
(iv) All the test to be conducted should be planned before implementing it  
(v) It follows pareto rule (80/20 rule) which states that 80% of errors comes from 20% of program components.  
(vi) Start testing with small parts and extend it to large parts.

### Types of Testing:-

#### 1. Unit Testing

It focuses on smallest unit of software design. In this we test an individual unit or group of inter related units. It is often done by programmer by using sample input and observing its corresponding outputs.  
Example:

a) In a program we are checking if loop, method or

function is working fine

b) Misunderstood or incorrect, arithmetic precedence.

c) Incorrect initialization

#### 2. Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components are combined to produce output.

Integration testing is of four types:

* 1. Top down
  2. Bottom up
  3. Sandwich
  4. Big-Bang

Example

(a) Black Box testing:- It is used for validation.

In this we ignores internal working mechanism and

focuses on **what is the output?**.

(b) White Box testing:- It is used for verification.

In this we focus on internal mechanism i.e.

**how the output is achieved?**

#### 3. Regression Testing

Every time new module is added leads to changes in program. This type of testing make sure that whole component works properly even after adding components to the complete program.  
Example

In school record suppose we have module staff, students

and finance combining these modules and checking if on

integration these module works fine is regression testing

4. **Smoke Testing**

This test is done to make sure that software under testing is ready or stable for further testing  
It is called smoke test as testing initial pass is done to check if it did not catch the fire or smoked in the initial switch on.  
Example:

If project has 2 modules so before going to module

make sure that module 1 works properly

#### 5. Alpha Testing

This is a type of validation testing.It is a type of acceptance testing which is done before the product is released to customers. It is typically done by QA people.  
Example:

When software testing is performed internally within

the organization

#### 6. Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for the limited number of users for testing in real time environment  
Example:

When software testing is performed for the limited

number of people

**7. System Testing**

In this software is tested such that it works fine for different operating system.It is covered under the black box testing technique. In this we just focus on required input and output without focusing on internal working.  
In this we have security testing, recovery testing , stress testing and performance testing  
Example:

This include functional as well as non functional

testing

#### 8. Stress Testing

In this we gives unfavorable conditions to the system and check how they perform in those condition.  
Example:

(a) Test cases that require maximum memory or other

resources are executed

(b) Test cases that may cause thrashing in a virtual

operating system

(c) Test cases that may cause excessive disk requirement

#### 9. Performance Testing

It is designed to test the run-time performance of software within the context of an integrated system.It is used to test speed and effectiveness of program.  
Example:

Checking number of processor cycles.

This article is contributed by **Kritka**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](http://www.contribute.geeksforgeeks.org/) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

**INTRODUCTION TO PROJECT**

**UNIVERSITY MANAGEMENT SYSTEM**

The University Management System is designed for any university to replace their existing manual paper based system. The project university Management system includes registration of students, storing their details into the system, and also computerized fee records as well as academic records of the students. The purpose of the project entitled as “UNIVERSITY MANAGEMENT SYSTEM” is to computerize the Front Office Management of University to develop software which is user friendly simple, fast, and cost – effective. The new system is to control the information of Students. Academic records, staff and courses availability as well as number faculty available for that particular course. I have designed the given proposed system in the JSP to automate the process of day to day activities of University like Academic record activities, Admission of New Student, Passed out students record, Assign a faculty , and finally compute the fees etc., All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.  These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks.

**Proposed University management system**  
  
“University Management System” is developed to computerize the following functions that are performed by the system:

1. Using this university management system user can give online registrations for the seat of the student.
2. Admin can register new student through this system.
3. Admin can also check graduate students Detail generated by the system.
4. This system is also keep academic as well as co-curricular activities record of students.
5. Admin can also check the faculty appointed to the class and vacancy of the faculty for any particular subject.
6. Administrator can add new faculty for the course and can also check all the faculty list of the university.
7. Admin manages the classes appointed for a particular semester students as well.
8. Labs availability is also checked by the admin.
9. Admin keeps a record of all the activities performed by the students and staff as well and generates the report accordingly.

**SRS of University Management System**

# Table of Contents

Contents

[Table of Contents 2](#_Toc18107347)

[1. Introduction 3](#_Toc18107348)

[1.1 Purpose 3](#_Toc18107349)

[1.2 DocumentConventions 3](#_Toc18107350)

[1.3 Intended Audience and ReadingSuggestions 3](#_Toc18107351)

[1.4 Project Scope 3](#_Toc18107352)

[1.5 References 4](#_Toc18107353)

[2. OverallDescription 5](#_Toc18107354)

[2.1 ProductPerspective 5](#_Toc18107355)

[2.2 ProductFeatures 5](#_Toc18107356)

[2.3 User Classes and Characteristics 6](#_Toc18107357)

[2.4 Operating Environment 6](#_Toc18107358)

[2.5 Design and ImplementationConstraints 6](#_Toc18107359)

[2.6 UserDocumentation 7](#_Toc18107360)

[2.7 Assumptions andDependencies 7](#_Toc18107361)

[3. External Interface Requirements 8](#_Toc18107362)

[3.1 UserInterfaces 8](#_Toc18107363)

[3.2 HardwareInterfaces 8](#_Toc18107364)

[3.3 SoftwareInterfaces 9](#_Toc18107365)

[3.4 Communications Interfaces 9](#_Toc18107366)

[4. Other NonfunctionalRequirements 10](#_Toc18107367)

[4.1 PerformanceRequirements 10](#_Toc18107368)

[4.2 Safety Requirements 10](#_Toc18107369)

[4.3 Security Requirements 10](#_Toc18107370)

[4.4 Software Quality Attributes 11](#_Toc18107371)

[Appendix A: Glossary 11](#_Toc18107372)

# 

**INTRODUCTION OF DOCUMENT**

**PURPOSE**

The main objective of this document is to illustrate the requirementsof the projectUniversity Management system.This documentdescribes the design decisions, architectural design and the detaileddesign needed to implement the system. It provides the visibility inthe design and provides information needed for software support. Thedocument gives the detailed description of the both functional andnon-functional requirements proposed by the client. The document isdeveloped after a number of consultations with the client andconsidering the complete requirement specifications of the givenProject. The final product of the team will be meeting therequirements of this document.

**DOCUMENT CONVENTIONS**

The following are the list of conventions and acronyms used in thisdocument and the project as well:

Administrator: A login id representing a user with user administrationprivileges to the software

* User:A general login id assigned to users.
* Client:Intended users for the software.
* SQL:Structured Query Language; used to retrieve informationfrom a database.
* SQL Server:A server used to store data in an organized format.
* ASP:Active Server Pages: A Web Page formatted on the serverand delivered to the browser.
* Layer:Represents a section of the project.
* User Interface Layer :The section of the assignment referring towhat the user interacts with directly.
* Application Logic Layer :The section of the assignment referringto the Web Server. This is where all computations are completed.
* Data Storage Layer :The section of the assignment referring towhere all data is recorded
* Data flow diagram :It shows the dataflow between the entities.
* Use Case: A broad level diagram of the project showing a basic overview.
* Boolean: A true/false notation.
* Interface: Something used to communicate across differentmediums.
* Unique Key: Used to differentiate entries in a database

**SCOPE**

Online Project Marking System is developing for School of Computing, University of Portsmouth and used to replace old paper work system and PUMS. OPMS is to build upon the existing web-based project marking system PUMS in order to implement the project marking process and allocating supervisor/ideas to students. This increase inefficiency of project marking, audit trails of marking process, give feedback to student, finally, publication and email student result. It provides a mechanism to edit the online marking form which makes the system is flexible

**OVERALL DESCRIPTION**

**PRODUCT PERSPECTIVE**

The proposed University Management System is an online University Management System. This System will provide a view, submit, online payment, uploading various documents and other resources. This view will be based on the categories like attendance view and daily activities. Further the University management staff personnel(faculty) can add/update/remove there sources or an automatic removal of accessing features when the time limit completes.

The System will also have an ADMIN who has full-fledged rights with regards to managing resources across branches – such as transferring books across these branches. The users can view, submit, online payment, uploading various documents and information about their account etc. there are basic two types of users on earth students and other are faculty members. Each user facilitates with a different account number having a profile along with a password for private use. The two types of users differ from each other due to the accessing limits to online University management system.

**PRODUCT FEATURES**

There are three different users who will be using this product:

University chancellor who will be acting as the administrator.

Faculty members who are second level users accessing UMS.

Student of the University who will be accessing the UMS online.

The features that are available to the Administrator are:

The administrator has the full-fledged rights over the UMS.

Can create/delete an account.

Can view the accounts.

Can change the password.

Can hide any kind of features from the both of users.

Insert/delete/edit the information of available on UMS.

Can access all the accounts of the faculty members/students

The features available to the Faculty members are:

Can mark the attendance of students online.

Can view the attendance online.

Can upload marks, assignments, reading materials forstudents.

The features available to the Students are:

Can view the different categories of assignments availablein their account.

Can view their marks.

Can view the various reading material.

Can view attendance.

Can view and modify its profile but can modify it to some limited range.

Can pay their fee online.

**USER CLASSES AND CHARACTERISTICS**

There are various kinds of users for the product. Usually web products are visited by various users for different reasons.

The users include:

Chancellor who will be acting as the controller and he will have all the privileges of administrator.

Faculty members who will be using the above features by accessing the UMS online.

Students who will be using the above features by accessing the UMS online.

**OPERATING ENVIRONMENT**

The product will be operating in windows environment. It will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

**DESIGN AND IMPLEMENTATION CONSTRAINTS**

The Product is developed using ASP. The backend database for this SQLServer. The product is accomplished with login facility so that specific function is available to specific student.

**USER DOCUMENTATION**

The product will include user manual. The user manual will include product overview, complete configuration of the used software (such as SQL server), technical details, backup procedure and contact information which will include email address. The products will be compatible with the Internet Explorer 6.0 or higher. The databases will be created in the Microsoft SQL server 2000.

**ASSUMPTIONS AND DEPENDENCIES**

The product needs following third party product:

Microsoft SQL server to store the database.

ASP to develop the Product.

**SYSTEM FEATURES**

**DATABASE-STORAGE**

**DESCRIPTION AND PRIORITY**

Proposed Database is intended to store, retrieve, update, andmanipulate information related to university which include

Profile of both users

Staff information

Student details

My account

Online payment

View attendance/marks/uploading of marks andassignments

**STIMULUS/RESPONSE SEQUENCES**

**RESPONSES FOR ADMINISTRATOR**

The administrator can Login and Logout. When the Administrator Logs into the University management system. The system will check for validity of login. If the Login and password are valid, the response to this action is the administrator will be able to modify, view, add, deleting and all other functions that can be performed on the database.

**FUNCTIONAL REQUIREMENTS**

This section gives the list of Functional and non-functional requirements which are applicable to the University Management System.

**INTERFACE REQUIREMENTS**

This section describes how the software interfaces with other software products or users for input or output.

**USER INTERFACES**

Describes how this product interfaces with the user.

**GUI**

Describes the graphical user interface if present. This section should include a set of screen dumps or mockups to illustrate user interface features.

Description

The user interface must be customizable by the administrator

Criticality

This issue is essential to the overall system. All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined.

Technical issues

In order to satisfy these requirements the design should be simple and all the different interfaces should follow a standard template. There will be the possibility of changing colors and images, plus switching between interfaces with the minimum impact for the users.

Risks

To reduce the circumstances under which this requirement might not able to be satisfied, all the designers must have been developed web sites previously and they must be aware of html restriction and cross browsers implementations before starting the designing. In order to reduce the probability of this occurrence the entire design team will be trained in basic html development and macromedia fireworks, this tool will be used instead of Photoshop.

Dependencies with other requirements

All user interfaces should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module.

**INPUT REQUIREMENTS**

**USER ACCESS**

Each faculty member and students assigned a unique identifier upon admission to the university. Both of them must know this. This identifying key maps to all his/her registration recording formation in the main registration system. Admitted and current students have their online registration accounts also enabled. Such account maybe disabled during his/her stay as matriculated student and/or after graduation or separation from the university.

**UPLOADING DATA**

Each faculty member should facilitates with uploading of data such assignments, their marks and other kind of reading material. Similarly such of option must be present their for students to upload their assignments.

**ONLINE PAYMENT**

The students should have the facility to pay their payment onlineany kind of university fee charges so as there should be facility to check whether the entered code for payment is a valid code ornot or in simple word a proper validation is required.

**NON-FUNTIONAL REQUIREMENTS**

**HARDWARE INTERFACES**

**SERVER SIDE:**

Operating System: Windows 9x/xp ,Windows ME

Processor: Pentium 3.0 GHz or higher

RAM: 256 Mb or more

Hard Drive: 10 GB or more

**CLIENT SIDE:**

Operating System: Windows 9x or above, MAC or UNIX.

Processor: Pentium III or 2.0 GHz or higher.

RAM: 256 Mb or more

**SOFTWARE INTERFACES**

Database: SQL Server.

Application: ASP (Active Server Pages)

Web Server: IIS (Internet Information Services (IIS) is a powerful Web server that provides a highly reliable, manageable, and scalable Web application infrastructure).

**COMMUNICATION INTERFACES**

The Customer must connect to the Internet to access the Website:

Dialup Modem of 52 kbps

Broadband Internet

Dialup or Broadband Connection with a Internet Provider.

**OTHER NON-FINCTIONAL REQUIREMENTS**

**PERFORMANCE REQUIREMENTS**

The proposed system that we are going to develop will be used as the Chief performance system within the different campuses of the university which interact with the university staff and students. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the university.

**SAFETY REQUIREMENTS**

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

**SECURITY REQUIREMENT**

We are going to develop a secured database for the university. There are different categories of users namely teaching administrator, Staff members and students etc. Depending upon the category of user the access rights are decided. It means if the user isan administrator then he can be able to modify the data, delete, append etc. All other users other than University Staff only have the rights to retrieve the information about database.

**SOFTWARE QUALITY ATTRIBUTES**

The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database.

**HARDWARE CONSTRAINTS**

The system requires a database in order to store persistent data. The database should have backup capabilities.

**SOFTWARE CONSTRAINTS**

The development of the system will be constrained by the availability of required software such as web servers, database and development tools.

**DESIGN CONSTRAINTS**

The system must be designed to allow web usability. That is, the system must be designed in such a way that will be easy to use and visible on most of the browsers.

## SOFTWARE QUALITY ATTRIBUTES

Flexibility, reusability, robustness, and maintainability of the HMS system should be maximized, in order for clients to be able to deploy custom settings of the FOSS HMS to their individual hospital network needs.

# 

# Appendix A: Glossary

##### 7Zip

* 1. Highly acclaimed and functional, multi-platform, FOSS compression file format standard
  2. Utilizes GNU LGPL License
  3. See <http://www.7-zip.org/>for more information

##### Baudrate

* 1. Rate of transfer of data over the internet/network
  2. Measured in bit per second(bps)

##### Bit

* 1. Binary Digit
  2. One zero(0) or one(1).

##### Database

* 1. Big memory address block which contains large set of data
  2. With subsets and fields that can search for by filter, read, and written

##### FOSS

* 1. **F**ree and **O**pen **S**ource **S**oftware
  2. Software that is freeware, and which has its source code available, for others to modify under the GNU Software License
  3. See<https://www.gnu.org/copyleft/gpl.html>for GPL information.

##### HIS

* 1. **H**ospital **I**nformation **S**ystem
  2. Element of health informatics
  3. Focuses mainly on the administration needs of hospitals
  4. In many implementations, a HIS is a comprehensive, integrated information system designed to manage all the aspects of a hospital's operation
     1. Medical
     2. Administrative
     3. Financial
     4. Legal issues
     5. Corresponding processing of services.
  5. (Paraphrase source:<https://en.wikipedia.org/wiki/Health_informatics>)

1. **HMS**
   1. **H**ospital **M**anagement **S**ystem

##### LAN

* 1. **L**ocal **A**rea **N**etwork
  2. Refers to computer connected to the internet/each other via physical Ethernet Cable.

##### Ribbon

* 1. Refers to the Windows GUI element that dynamically changes the items in a menu bar

1. **SRS**
   1. **S**oftware **R**equirements **S**tatement
   2. Statement clarifying the what a software project is supposed to be engineered to do
   3. Specifies the limits, constraints, and big-picture, abstract plan of the software engineering

##### SQL

* 1. **Structured Query Language**

##### W7 RibbonStyle

* 1. A GUI style in the HIS software which looks and acts like the ribbon in Microsoft Office2007

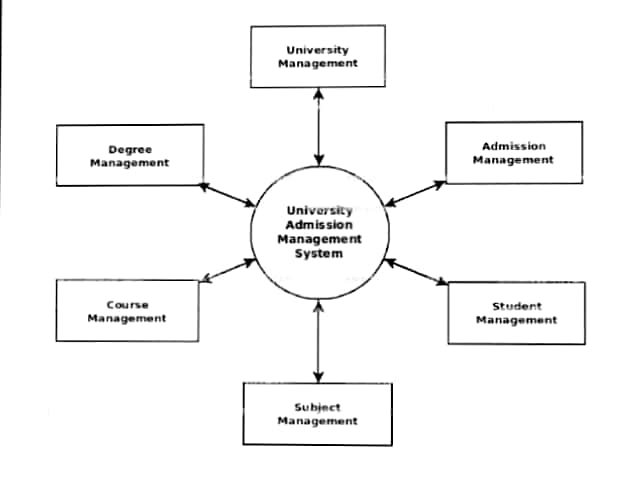
##### Windows API

* 1. Windows **A**pplication **P**rogramming **I**nterface

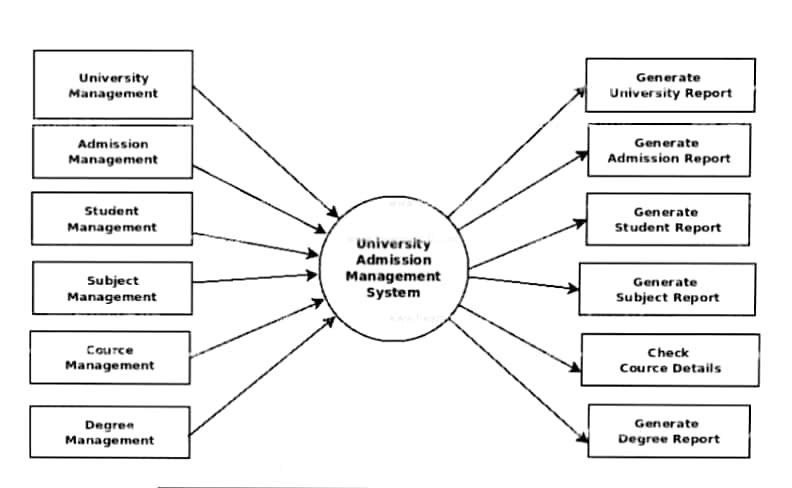
**DATA FLOW DIAGRAMS (DFD’s)**

**DFD**:- Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

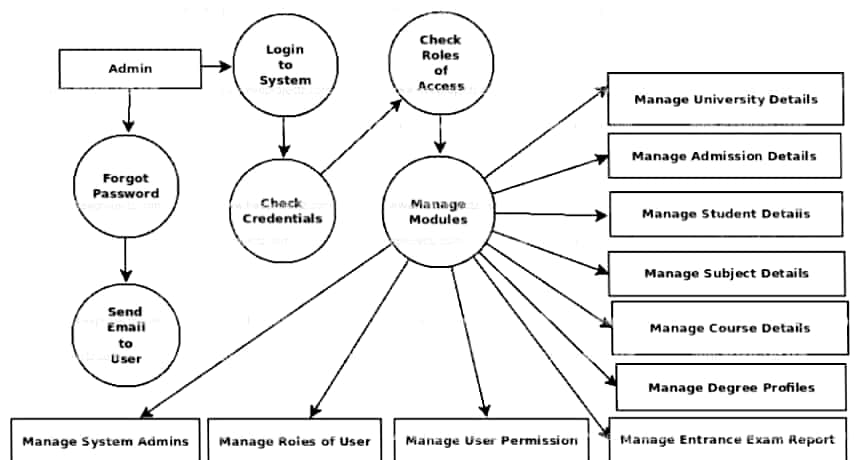
Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.



**0 LEVEL DFD**



**1 LEVEL DFD**

****

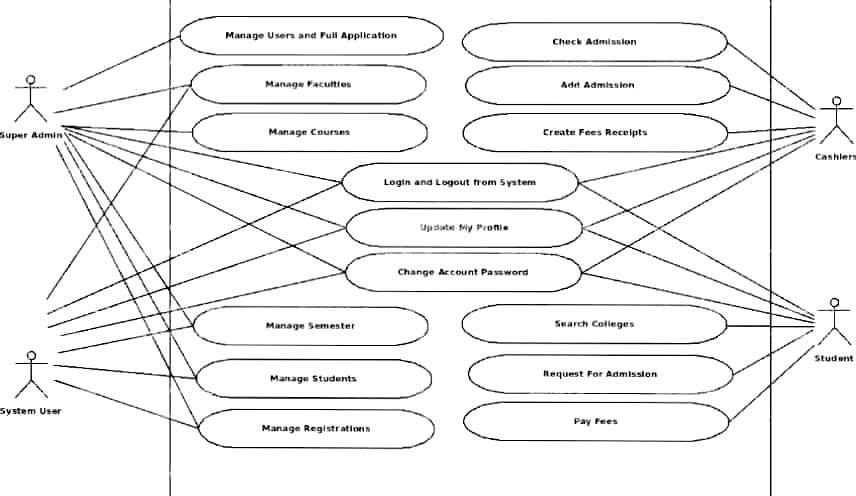
**2 LEVEL DFD**

**UNIFIED MODELING LANGUAGE(UML)**

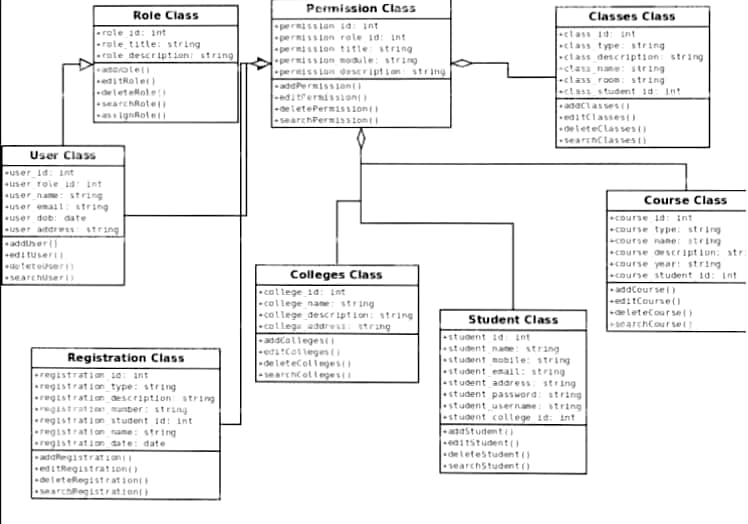
**UML**:- Unified Modeling Language (UML) is a general purpose modeling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language, it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modeling, design and analysis. The Object Management Group (OMG) adopted Unified Modeling Language as a standard in 1997. Its been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

**USECASE DIAGRAM**: -A use case diagram is a dynamic or behavior diagram in UML**.**Use case diagrams model the functionality of a systemusing actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform

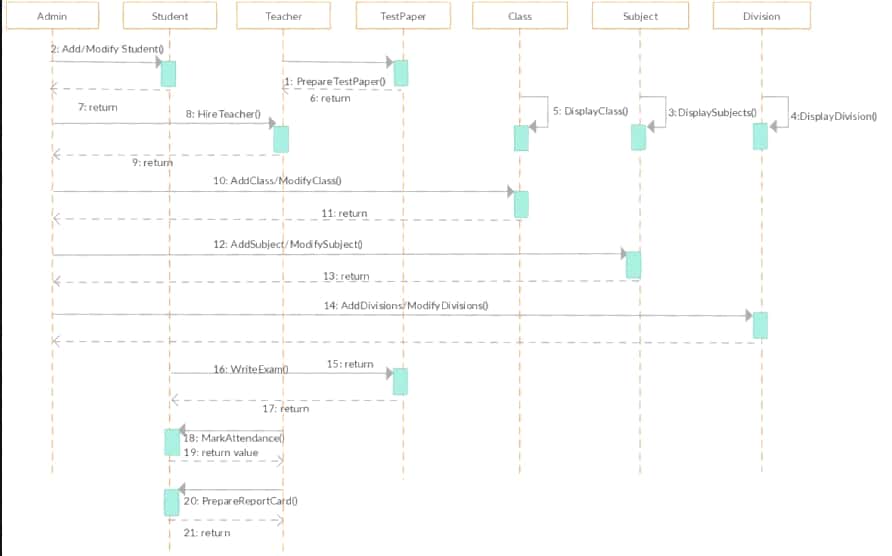
 **USE CASE DIAGRAM**

**CLASS DIAGRAM**: -In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects



**CLASS DIAGRAM**

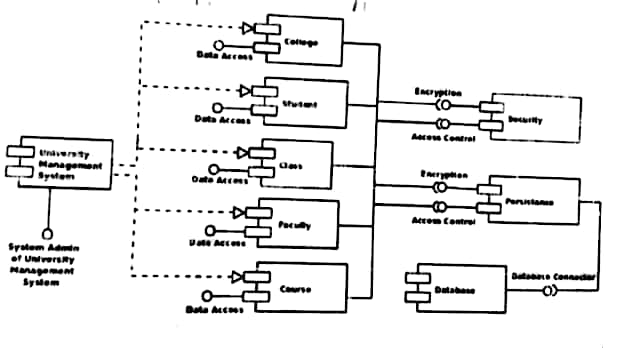
**SEQUENCE DIAGRAM**: -A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

****

**SEQUENCE DIAGRAM**

**COMPONENT DIAGRAM**:-

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.



**COMPONENT DIAGRAM**

### Unit Testing

Unit testing is the practice of testing small pieces of code, typically individual functions, alone and isolated. If your test uses some external resource, like the network or a database, it’s not a unit test.

Unit tests should be fairly simple to write. A unit tests should essentially just give the function that’s tested some inputs, and then check what the function outputs is correct. In practice this can vary, because if your code is poorly designed, writing unit tests can be difficult. Because of that, unit testing is the only testing method which also helps you write better code – Code that’s hard to unit test usually has poor design.

### Integration Testing

As the name suggests, in integration testing the idea is to test how parts of the system work together – the integration of the parts. Integration tests are similar to unit tests, but there’s one big difference: while unit tests are isolated from other components, integration tests are not. For example, a unit test for database access code would not talk to a real database, but an integration test would.

Integration testing is mainly useful for situations where unit testing is not enough. Sometimes you need to have tests to verify that two separate systems – like a database and your app – work together correctly, and that calls for an integration test. As a result, when validating integration test results, you could for example validate a database related test by querying the database to check the database state is correct.

