**SOFTWARE ENGINEERING PROJECT**

**INTERNAL ASSESSMENT EVALUATION SYSTEM**

**Presented By:**

Nikhil (13CS2532)

**INDEX**

1. INTRODUCTION
   1. Overview
   2. Disadvantages of Manual System
   3. Objective of the project
   4. Scope of the project
2. SOFTWARE PROJECT MANAGEMENT
   1. Introduction
   2. Project plan

2.2.1 Software Process Model

2.2.2 Team Structure

2.2.3 Risk Analysis and Management

2.2.4 Software Cost Estimation

1. REQUIREMENT ANALYSIS AND SPECIFICATION
   1. Introduction
   2. SRS Document
   3. Structured Analysis

3.3.1 Entity Relationship Diagram

3.3.2 Data Dictionary

3.3.3 Data Flow Diagram

1. SOFTWARE DESIGN
   1. Introduction
   2. Data Design
   3. Architectural Design
   4. Interface Design
   5. Pseudo code
2. SOFTWARE TESTING
   1. Software Testing Strategies
   2. Software Testing Techniques
   3. Sample Unit Testing
   4. Cyclomatic Complexity

BIBLIOGRAPHY

**CHAPTER 1**

**INTRODUCTION**

* 1. **OVERVIEW**

The performance of a student cannot be judged, however hard a student may attempt, during the stipulated period of 3 hours in the final exam. Hence, **Delhi University** has started a system of internal assessment. In this system, in each paper, **25 marks are** to be awarded to the college students on the basis of their individual performance throughout the semester. The marking scheme of the **INTERNAL ASSESSMENT EVALUATION SYSTEM** is grouped in the following 3 categories

1. **10 marks** for **test**
2. **5 marks** for the **attendance**
3. **10 marks** for the **assignments and projects** submitted by each student.

Currently, in most of the colleges internal assessment marks are maintained manually.

**1.2 DISADVANTAGES OF THE MANUAL SYSTEM**

1. Maintaining records as paperwork is a cumbersome task.
2. Too many calculations done manually, lead to chances of errors which can in turn disrupt the final outcome.
3. There can be threat to the security of the records, since anyone can easily access and modify these.

**1.3 OBJECTIVE OF THE PROJECT**

The goal of this project is to create an automated system for internal assessment for three year undergraduate students of Delhi University. This software will capture attendance, project/assignment marks, test marks and calculate final marks.  The software will be a multiuser system. The expected users of this system will be students, faculty, and admin of the college. The software should enable the students to view their attendance and internal marks in each subject. Faculty should be able to enter attendance and marks of the students they are teaching. Admin should be able to generate different report of internal assessment class-wise. This software will enhance the security features (by using passwords) that are void in the traditional ways of implementation of the information storage.

* 1. **SCOPE OF THE PROJECT**

The software product, **INTERNAL ASSESSMENT EVALUATION SYSTEM** will be a reporting application that will be used for calculating the internal assessment of students.

Each user of the system is given a username and password. The user is allowed to access the software only if they enter the correct password. This will provide **security from unauthentic users**.

* Each teacher marks daily **attendance** and at the end of each month, students attendance should be updated in the software. At the end of semester marks not amounting to more than **5** are awarded to each student, depending on the percentage of lectures attended by each, to the total lectures. Students having above 90% attendance are awarded 5 marks , above 85% awarded 4 marks, above 80% 3 marks, above 75% are awarded 2 marks and above 70% are awarded 1 mark.
* The students submit their **assignments/project** periodically, which are corrected by the teacher concerned. When the semester ends, marks amounting to not more than **10** are awarded to the student keeping in view his/her performance.
* Similarly, marks obtained in the **class tests** are taken into consideration, and on the basis of actual performance, each student is awarded marks, at the close of the session, which don’t exceed **10.**

**CHAPTER 2**

**SOFTWARE**

**PROJECT**

**MANAGEMENT**

**2.1 INTRODUCTION**

**Project management** involves planning, monitoring and control of people, processes and events that occur, as software evolves from a preliminary concept to an operational implementation. Effective software project management focuses on the 4 P’s:-

**P**eople**, P**roduct, **P**rocess, **P**roject.

**THE PEOPLE**

Software engineering institute has developed a people management capability maturity model (**PM-CMM**). The people management maturity model defines the key practice areas [KPA’s] for software people like recruiting, selection, performance management, training, compensation, carrier development, organization and work design and team / culture development.

**THE PRODUCT**

Before a project is planned, product objectives and scope should be established, alternative solutions should be considered and technical and management constraints should be identified.

**Objectives** identify the overall goal of the product from customer’s point.

**Scope** identifies the primary data, functions and behaviors that characterize the product.

**Alternatives** enable managers to select the best approach given constraints imposed by technical interfaces, personnel availability, delivery deadlines and budgetary restrictions.

Thus the product factor helps to define the accurate cost estimation, effective risk assessment and a manageable project schedule.

**THE PROCESS**

A software process provides the framework from which a comprehensive plan for software development can be established. **Framework activities** are populated with tasks, milestones, work products and quality assurance points. These activities characterize the software product and the project team. **Umbrella activities** i.e. software quality assurance, software configuration management and measurement overlay the process model.

**THE PROJECT**

Planned and controlled software projects are conducted to manage complex. To avoid project failure, the project manager must avoid a set of common warning signs, understand critical success factors and develop a common sense approach for planning, monitoring and controlling the project.

**2.2 PROJECT – PLAN**

**2.2.1 SOFTWARE PROCESS MODEL**

To solve a particular problem, the project team must incorporate a development strategy that encompasses the process, methods and tools. This strategy is often referred to as a process model or a “**SOFTWARE ENGINEERING PARADIGM**”. The use of a particular process model or software paradigm is based on the nature of the application. The following points state the need of a particular software paradigm for development of software.

* To improve the quality of software.
* To increase the productivity of software development.
* To develop software on time.
* To produce a reliable software.
* To develop the software with in cost estimates.

The software process model used is **LINEAR SEQUENTIAL MODEL / WATERFALL MODEL** because of the following reasons:

* This is small software which demands a systematic and sequential approach to software development.
* The project is dedicated to perform only the given set of tasks and any diversion from the same will not vary by a significant margin for a substantial period of time.
* The structure of the software is pretty straight-forward with a simple hierarchy and without many complex transfers of control.

**Waterfall Model / Linear Sequential Model**

This is sometimes called the **Classic Life Cycle** or **Linear Sequential Model**. It suggests a systematic approach to software development that begins at the system level and progress through analysis, design, coding, testing and support.

**DESIGN**

**ANALYSIS**

**CODE**

**TEST**

The following are the activities that the Linear Sequential Model applies:-

**System/Information engineering and modeling**

It is essential when software must interact with other elements such as hardware, people and database. System engineering and analysis encompass requirement gathering at the system level with a small amount of top-level design and analysis. Information engineering encompass requirement gathering at the strategic business level and at the business area level.

**Software Requirement Analysis**

It is a necessary step to understand the nature of the problem to be built. This phase gathers the input, output, etc. Requirement for both the system and the software are documented leading to the requirement specification report.

**Design**

This phase focuses on the software architecture, data structures, tables, flow diagrams, interface representations and procedural details. The design translates requirements into a presentation of software that can be assessed and reviewed before code generation begins.

**Code Generation**

The design developed above has to be translated into a machine-readable form. The code generation step performs this task.

**Testing**

After the code has been generated, program testing begins. Testing is done to uncover errors and ensure that defined input produces the actual results as required by the user.

**Support**

This is a phase when software will undoubtedly undergo change after it is delivered to the customer. Change will occur because errors have been encountered, because the software must be adapted to accommodate changes in its external environment, or because the customer requires functional or performance enhancements. Software support/maintenance reapplies each of the preceding phases to an existing program rather than a new one.

* + 1. **TEAM STRUCTURE**

We use **democratic decentralized** [**DD**] team structure in our project. Our team comprises of three members:

**Ankita Dagar**

**Nikhil Jangra**

**Shubham Saini**

**Advantages**

* Generate better solutions.
* Have greater probability of success when working on difficult problems.
* Best applied to programs with low modularity because of the higher volume of communication needed.
* Results in high morale.

**2.2.3 RISK ANALYSIS & MANAGEMENT**

**Risk** always involves two characteristics:-

UNCERTAINITY

LOSS

**Risk analysis and management** is a series of steps that help a software team to understand and manage uncertainty. Many problems can plague a software project. A risk is a potential problem-it might happen, it might not. But regardless of the outcome, it’s really a good idea to identify it, assess its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur.

**Types of risk**

**PROJECT RISK**

They identify potential budgetary, schedule, personnel, resource, custom potential and requirements problem and their impact on software project. They threaten the project plan.

**TECHNICAL RISK**

They identify potential design, implementation, interface verification, and maintenance problem. They threaten the quality and timeliness of software to be produced.

**BUSINESS RISK**

They often jeopardize the project or the product and include market risk, strategic risk, management risk and budget risk.

**Risk strategies**

**REACTIVE**

A reactive strategy monitors the risk project for likely risk and set aside resources to deal with them, should they become actual problems. Software team does nothing about risks until something goes wrong.

**PROACTIVE**

A proactive strategy begins long before technical work is initiated. Potential risks are identified, their probability impact is assessed, and they are ranked by importance.

**Risk analysis**

**Risk analysis** is a technique to identify and assess factors that jeopardize the success of a project or achieving a goal. This technique also helps define preventive measures to reduce the probability of these factors from occurring and identify counter measures to successfully deal with these constraints when they develop to avert possible negative effects on the competitiveness of the company.

This is achieved by:-

* Risk avoidance
* Risk monitoring
* Risk management and contingency plan

**RMMM PLAN (Risk Mitigation, Monitoring and Management Plan)**

It documents all work performed as a part of risk analysis and is used by project manager as a part of overall project plan. Once RMMM has been documented and the project has begun, risk mitigation and monitoring steps commence.

**Risk management**

Following steps can be taken for resolution of the mentioned risks:

* Try to develop healthy communication with clients’ staff so as to easily gather requirements and to train and guide them about the software.
* Divide the work among team members properly to meet the deadlines.
* Try to finish the work at least 10 days before the deadline, as many changes have to be incorporated after that.
* Timely check the space availability and size of the software.
* Take client approvals after each step of project development.
  + 1. **SOFTWARE COST ESTIMATION**

Software Cost Estimation is very important for a software organization. Overestimated cost causes the management to disapprove the project whereas underestimates may lead to failure of the project by exceeding the budget. Estimating the product size is fundamental to estimate the efforts, time and cost of a planned software project. There are a wide range of models that can be to access, predict and control software costs. One of the methods used to estimate the size of code is Function Points.

**Function Point**

The basic idea behind the function point analysis is that the size of a software product is directly dependent on the number and type of different functions it performs. Function points are computed by first calculating an unadjusted function point count (UFC). Counts are based on the following five parameters:

* Number of User Inputs (EI)
* Number of Outputs (EO)
* Number of files (ILF)
* Number of Interfaces (EIF)
* Number of User Queries (EQ)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MEASUREMENT PARAMETER** | **COUNT** | **SIMPLE** | **AVERAGE** | **COMPLEX** | **TOTAL** |
| EI | 50 | 3 | 4 | 6 | 150 |
| EO | 20 | 4 | 5 | 6 | 100 |
| ILF | 12 | 7 | 10 | 15 | 120 |
| EIF | 20 | 5 | 7 | 10 | 140 |
| EQ | 20 | 3 | 4 | 6 | 80 |
| **TOTAL** | | | | | 590 |

**Complexity Factors**

1. Reliable Backup and recovery - 5
2. Data Communication - 4
3. Distributed processing function - 0
4. Performance - 4
5. Heavily used - 3
6. Online data entry - 5
7. Operational ease - 4
8. Online update - 5
9. Complex interface - 2
10. Complex processing - 3
11. Reusability - 4
12. Installation ease - 1
13. Multiple sites - 4
14. Facilitate change - 3

Σ fi= 48

TCF = 0.65 + 0.01\* (Σ fi ) = 0.65 + 0.01\*48 = 0.65 + 0.48 = 1.13

FP = UFC \* TCF = 590\*1.13 = 666.7

**2.2.5 TIME - LINE CHART**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO.** | **TASK** | **DATE OF START** | **DATE OF END** |
|  | REQUIREMENT GATHERING AND ANALYSIS | 15/2/2015 | 25/2/2015 |
|  | DESIGN |  |  |
| 2.1 | Data design | 26/2/2015 | 28/2/2012 |
| 2.2 | Architectural design | 1/3/2015 | 7/3/2015 |
| 2.3 | Interface design | 8/3/2015 | 20/3/2015 |
| 2.4 | Pseudo code | 21/3/2015 | 31/3/2015 |

**CHAPTER 3**

**REQUIREMENT**

**ANALYSIS**

**AND**

**SPECIFICATION**

**3.1 INTRODUCTION**

**Requirement analysis** is a software engineering task that bridges the gap between system level requirements engineering and software design.

The software requirements analysis may be divided into five areas of efforts:-

**Problem recognition**

Recognition of basic problem elements as perceived by the users.

**Evaluation and synthesis**

Define all data objects, evaluate the flow and content of information, define and elaborate all functions, understand software behavior and establish interface characteristics.

**Modeling**

Functional models represent the information that software transforms, functions enabling the transformation, and behavior of the system during transformation.

**Specification**

State the goals and objectives of the software describing it in context of the computer based system.

**Review**

Changes to the specification may be recommended.

**Analysis Principles**

* The information domain of a problem must be represented and understood.
* The functions to be performed by software must be defined.
* The behavior of the software must be represented.
* The models that depict information function and behavior must be partitioned in a manner that uncovers detail in a layered fashion.
* The analysis process should move from essential information towards implementation detail.

**3.2 SRS Document**

# Introduction

The following subsections of the Software Requirements Specifications (SRS) document provide an overview of the entire SRS.

## Purpose

The Software Requirements Specification (SRS) will provide a detailed description of the requirements for the Internal Assessment Evaluation System (IAES). This SRS will allow for a complete understanding of what is to be expected of the IAES to be constructed. The clear understanding of the IAES and its’ functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. This SRS will provide the foundation for the project. From this SRS, the IAES can be designed, constructed, and finally tested.

## Scope

The software product to be produced is an Internal Assessment Evaluation System which will automate the internal assessment procedure. The first subsystem is a Master-Record updation that will do all the master database updation i.e. filling in details of students, faculty, courses, mapping of faculty and subjects. The second subsystem is Attendance System which keeps track of student’s attendance month-wise. The third subsystem is an Internal Assessment. These three subsystems’ functionality will be described in detail in section 2-Overall Description.

There are three end users for the IAES. The end users are Admin, faculty and Students. Admin should be able to make changes into master tables, generate reports of Student’s attendance and internal assessment. Faculty should be able to update student’s attendance at the end of each month. At the end of each semester faculty should be able to enter class-test/project presentation marks of each student. Student should be allowed to view his/her attendance and internal assessment marks.

The Internal Assessment Evaluation System’s objective is to provide a system to calculate the internal assessment marks of each student at the end of the semester. Without automation the calculation of internal assessment marks has become an unwieldy task. The calculation of internal assessment marks will be simplified by a considerable amount through the automated system. The system will be able to handle many services to take care of all queries in a quick manner. The system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction.

## Definitions, Acronyms and Abbreviations

SRS – Software Requirements Specification

IAES – Internal Assessment Evaluation System

Subjective satisfaction – The overall satisfaction of the system

End users – The people who will be actually using the system

## Overview

The SRS is organized into two main sections. The first is The Overall Description and the second is the Specific Requirements. The Overall Description will describe the requirements of the IAES from a general high level perspective. The Specific Requirements section will describe in detail the requirements of the system.

# The Overall Description

It describes the general factors that affect the product and its requirements. This section does not state specific requirements, instead it provides a background for those requirements, which are defined in section 3, and makes them easier to understand.

## Product Perspective

The IAES is a web based application. It is totally self-contained.

### Hardware Interfaces

The IAES will be placed on website of the college.

### Software Interfaces

All databases for the IAES will be configured using Oracle 8i. These databases include information about courses, students and faculty. These can be modified by the end users. The course database will include different courses run by the college, different subjects offered to students in each semester. The faculty database will include information about faculty members of the college and the different subjects taught by them in the current semester. The student database will contain all the personal information about student, course they are enrolled in, no. of classes attended by student in each subject month-wise, marks obtained by student in assignment and class-test.

## Product Functions

Master Data Updation

* Allows for entering information about different courses running in the college.
* Allows for entering information about the faculty and students.

Attendance System

* Allows the faculty to enter the attendance of his/her students month-wise.
* Should be able to generate the list of students having short attendance at the end of every semester.

Internal Assessment Evaluation System

* Allows the faculty to enter marks of class-test and project/assignment.
* Generate internal assessment of each student at the end of semester.

## Assumptions and Dependencies

## The system is not required to save generated reports.

# Specific Requirements

This section contains all the software requirements at a level of detail, that when combined with the system context diagram, use cases, and use case descriptions, is sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

## External Interfaces

The Internal Assessment Evaluation System will use the standard input/output devices for a personal computer. This includes the following:

* Keyboard
* Mouse
* Monitor
* Printer

### User Interfaces

The User Interface Screens are described in table 1.

Table 1: Internal Assessment User Interface Screens

| **Screen Name** | **Description** |
| --- | --- |
| Login | User id, Password, Type Log into the system as a Faculty, Admin or Student |
| Faculty | IA Entry, Reports, Change Password, Signout |
| Attendance | Screen for entering attendance of students  Course, Sem, Subject, month, List of students,  For each student max attendance, classes attended |
| Class test | Screen for entering class test marks of students  Course, Sem, Subject, List of students,  For each student max marks, marks obtained |
| Assignment Marks | Screen for entering assignment marks of students  Course, Sem, Subject, List of students,  For each student max marks, marks obtained |
| Students Record | Add or modify student records |
| Course Record | Add or update course records |
| Faculty record | Add or update faculty records |
| Subject record | Add or update subject records |
| Faculty Subject | Associate faculty with subject |
| Reports | Select, view, save, and delete reports |

### Software Interfaces

The system shall interface with an Oracle or Access database.

### Hardware Interfaces

The system shall run on any explorer.

### Communication Interfaces

The system shall be a standalone product that does not require any communication interfaces.

## Functional Requirements

Functional requirements define the fundamental actions that system must perform. The functional requirements for the system are divided into three main categories, Admin, Faculty and Students. For further details, refer to the use cases.

1. **Admin**
   1. The system shall record information about all courses.
   2. The system shall record information about all subjects/paper in different courses semester-wise.
   3. The system shall record information about all the faculty teaching in the college.
   4. The system shall record information about all the students enrolled in different courses.
   5. The system shall record information about which faculty is taking which subject.
   6. Admin shall be able to generate report of internal assessment class-wise.
   7. Admin shall be able to generate a list of student short of attendance.
   8. Admin shall be able to generate a list of paper along with the name of the faculty teaching that paper.
2. **Faculty**
   1. Each faculty shall be allowed to enter the attendance of all students in his/her subject month-wise.
   2. Each faculty should be able enter the marks of all students in his/her subject at the end of each semester.
   3. Each faculty should be allowed to generate the report of attendance and internal assessment mark of his/her subject.
3. **Student**
   1. Students should be able to view their marks and attendance.
   2. Students should not be allowed to modify their marks and attendance.

## Nonfunctional Requirements

Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

### Performance Requirements

Performance requirements define acceptable response times for system functionality.

* The load time for user interface screens shall take no longer than two seconds.
* The log in information shall be verified within five seconds.
* Queries shall return results within five seconds.

### Logical Database Requirements

The logical database requirements include the retention of the following data elements. This list is not a complete list and is designed as a starting point for development.

**USER INFORMATION**

The security of the software will be maintained with the following inputs:

User Name

User ID

Password

Type: (Admin / faculty / student)

**COURSE**

The course information is maintained as follows:

Course Code

Course Name

**SUBJECT**

This record contains following fields:

Subject Name

Subject Code

**FACULTY INFORMATION**

The faculty information includes:

Faculty name

Faculty code

**DATABASE OF THE STUDENTS**

The database of each student is inclusive of:

College Roll no.

University roll no.

Student’s name

Birth date

Father’s name

Mother’s name

Address

Phone no.

**STUDENTS ATTENDANCE**

The attendance record will contain the following fields:

College Roll no.

Subject code

**T**otal lectures

**L**ectures attended

**ASSIGNMENTS / PROJECT**

Following are the fields to be included in this record:

Subject code

College Roll no.

Maximum marks

Marks scored

**CLASS TEST MARKS**

The class test marks record will contain the following:

Subject code

College Roll no.

Maximum marks

Marks scored

**TOTAL INTERNAL ASSESSMENT MARKS**

This is the final record including:

Subject code

College Roll no.

Attendance marks

Assignment/project marks

Class test marks

Total marks

### Design Constraints

The Internal Assessment Evaluation System shall be a web-based application. The system shall be developed using Java and an Access or Oracle database.

### Standards Compliance

There shall be consistency in variable names within the system. The graphical user interface shall have a consistent look and feel.

### Reliability

Specify the factors required to establish the required reliability of the software system at time of delivery.

### Availability

The system shall be available around the clock.

### Security

Each user will be able to log in to the Internal Assessment Evaluation System through a username and password. Each user is assigned a type. Based on the type of user each user is given access to specific data only. Students are allowed to view their marks only. Faculties are allowed to modify marks of their subjects only.

### Maintainability

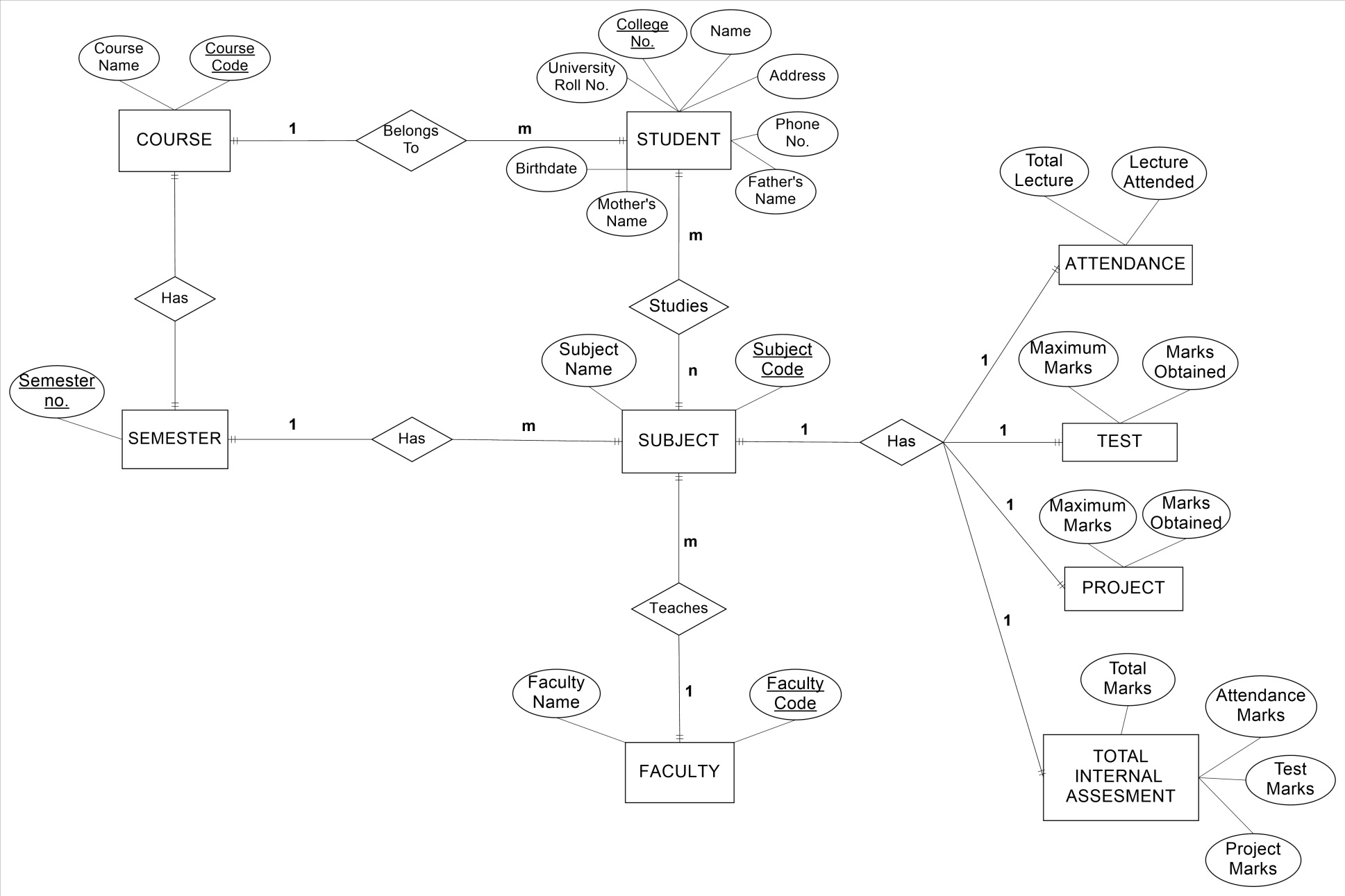
The Internal Assessment Evaluation System is being developed in Java. Java is an object oriented programming language and shall be easy to maintain.

### Portability

The Internal Assessment Evaluation System shall run in any browser.

## 3.3 STRUCTURED ANALYSIS

### 3.3.1 ENTITY RELATIONSHIP DIAGRAM

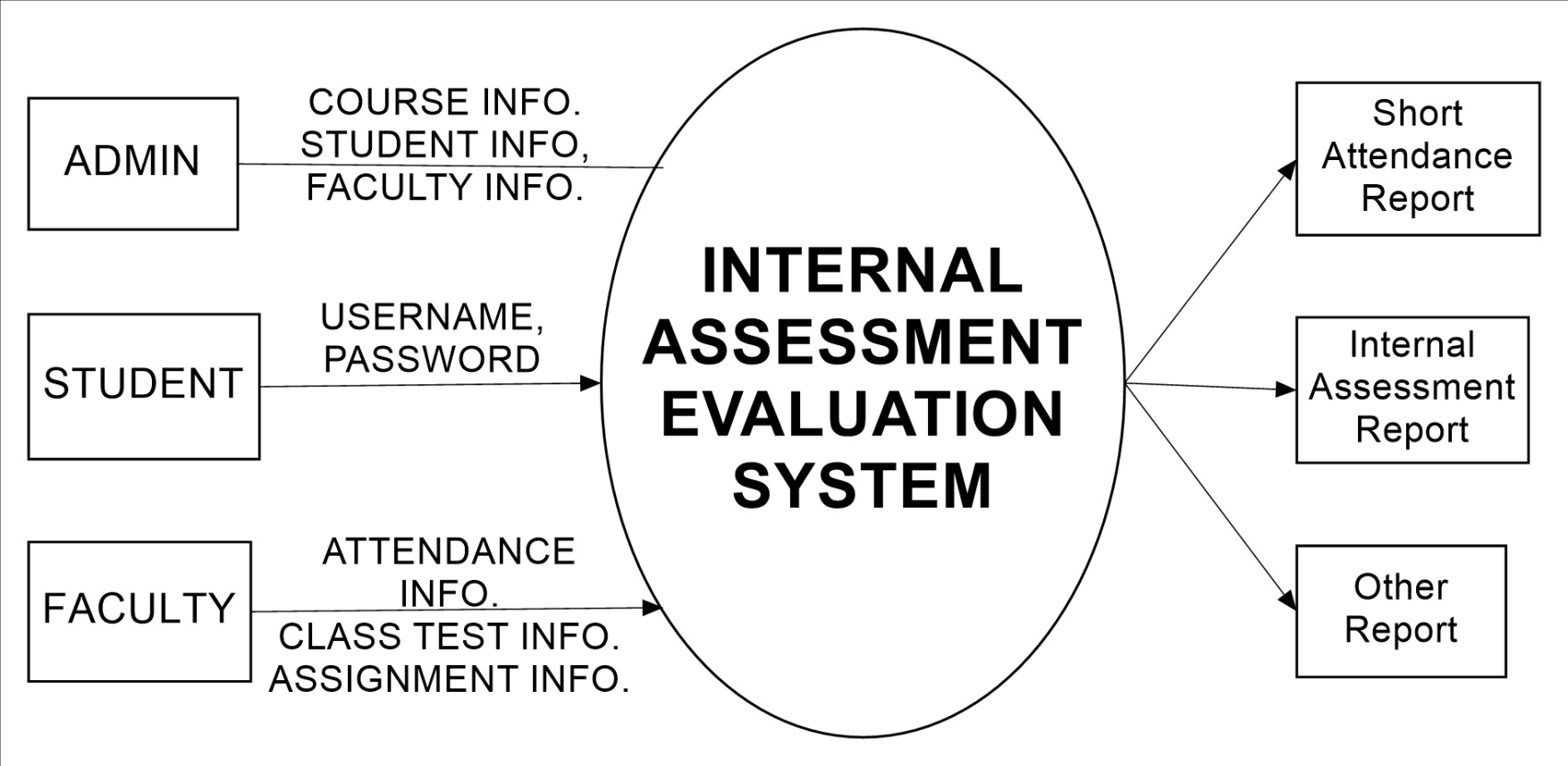


### DATA DICTIONARY

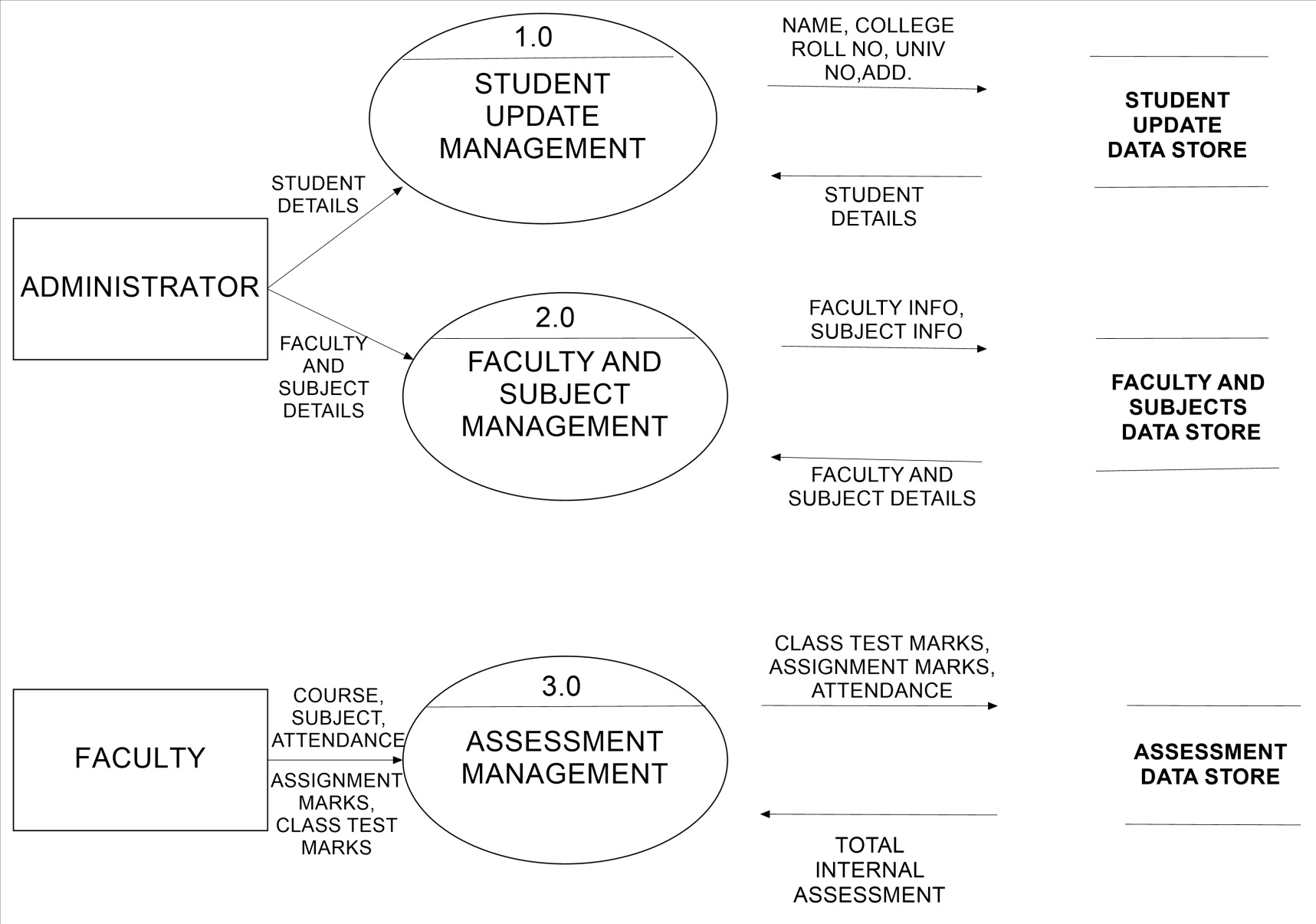
|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Field** | **Data Type** | **Purpose** |
| 1 | User Name | Char | Valid user name to enter the system |
| 2 | User ID | Char | Valid ID code |
| 3 | Password | Alphanumeric | Valid password to enter in the system |
| 4 | Type: (Admin / faculty / student) | Char | Mode of entering the system |
| 5 | Course Code | Numeric | Uniquely identifies the course |
| 6 | Course Name | Char | Name of the course |
| 7 | Subject Name | Char | Name of the subject |
| 8 | Subject Code | Numeric | Uniquely identifies the subject |
| 9 | Faculty name | Char | Name of the faculty members |
| 10 | Faculty code | Numeric | Uniquely identifies the faculty members |
| 11 | College Roll no | Alphanumeric | Uniquely identifies each student in the college |
| 12 | University roll no. | Numeric | Uniquely identifies each student in the university |
| 13 | Student’s name | Char | Name of the student |
| 14 | Semester no | Numeric | Current semester number |
| 15 | Birth date | Date | Date of birth of student |
| 16 | Father’s name | Char | Name of student’s father |
| 17 | Mother’s name | Char | Name of student’s mother |
| 18 | Address | Alphanumeric | Address of student |
| 19 | Phone no. | Numeric | Phone number of student |
| 20 | Total lectures | Numeric | Total lectures held in the semester |
| 21 | Lectures attended | Numeric | Number of lectures attended by student |
| 22 | Maximum marks (assignment) | Numeric | Total marks of the assignment |
| 23 | Marks scored | Numeric | Marks scored in assignment |
| 24 | Maximum marks (class test) | Numeric | Total marks of class test |
| 25 | Marks scored | Numeric | Marks scored in class test |
| 26 | Attendance marks (out of 5) | Numeric | Marks of attendance on the basis of number of lectures attended |
| 27 | Assignment/project marks (out of 10) | Numeric | Total marks scored by student in assignment out of 10 |
| 28 | Class test marks (out of 10) | Numeric | Total marks scored by student in class tests out of 10 |
| 29 | Total marks | Numeric | Total marks in internal assessment |

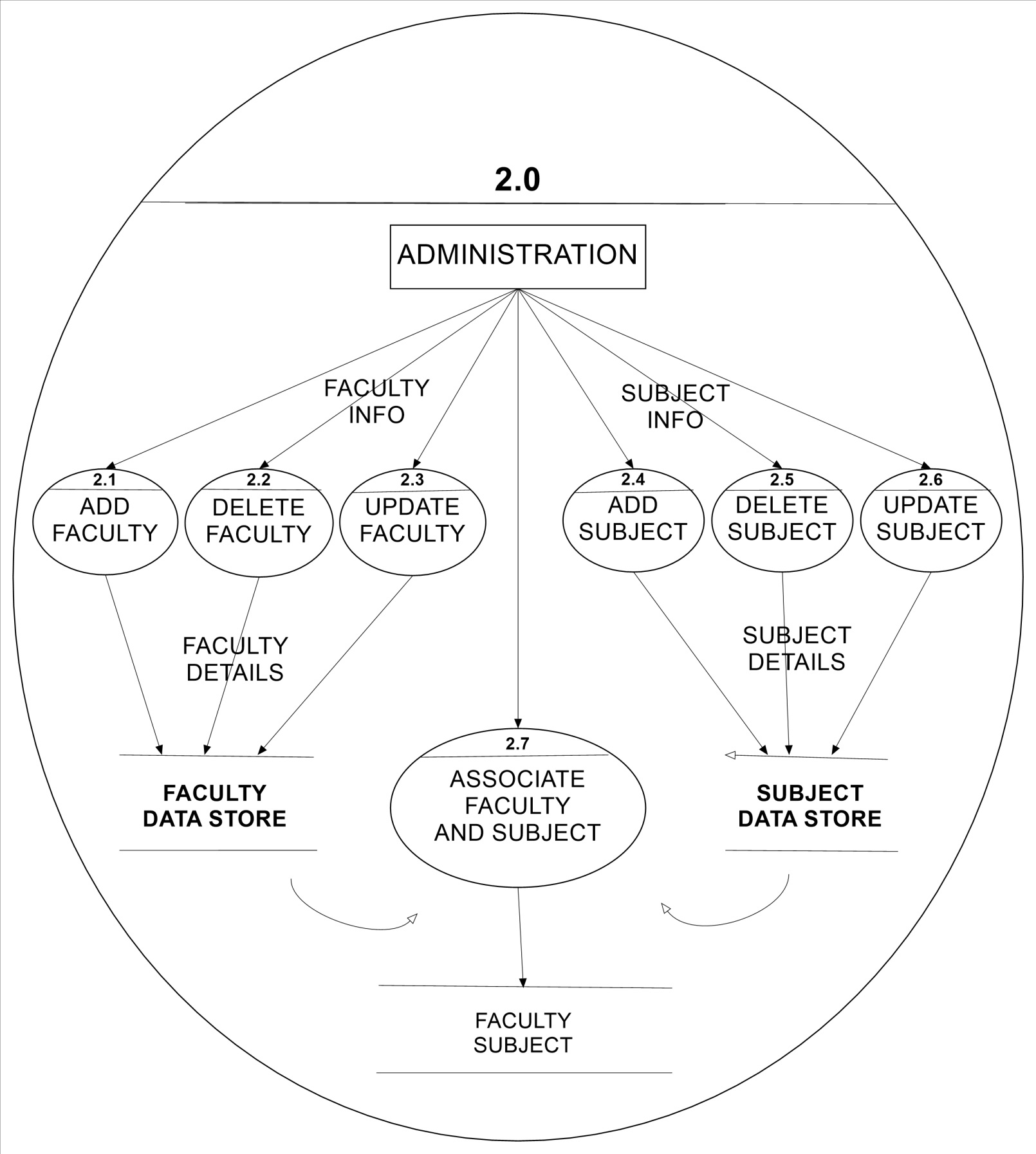
**3.3.3 DATA FLOW DIAGRAM**

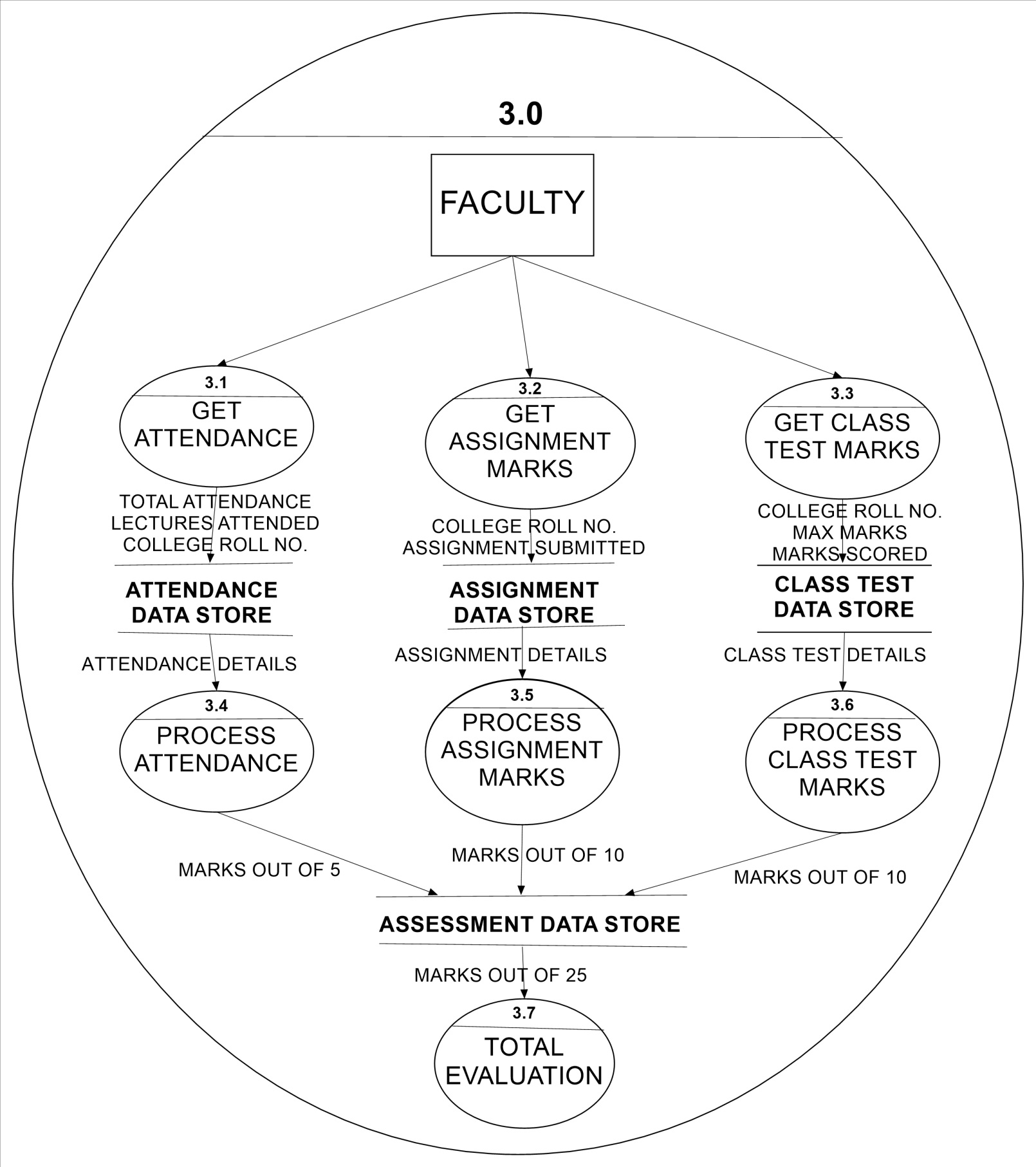
**CONTEXT LEVEL DIAGRAM**



**LEVEL 1 DFD**

******

**LEVEL 2 DFD****

****

**Chapter 4**

**SOFTWARE**

**DESIGN**

## INTRODUCTION

Design phase of the software development deals with transforming the requirements of the client into a form implement able using a programming language.

Software design is applied regardless of the software process model that is used. Beginning once software requirements have been analyzed and specifies, software design is the first of three technical activities—design, code generation, tests that are required to build and verify the software.

A good software design is a series of step-by-step procedures to do the desired act.

Design task comprises of:--

**Data Design**

It transforms the information domain model created during analysis into the data structures that will be required to implement the software.

**Architectural Design**

It defines the relationship between major structural elements of the software.

**Interface Design**

It describes how the software communicates within itself, with systems that interoperate with it and with the users who use it.

**Component Level Design**

It transforms structural elements of software architecture into a procedural description of software components.

**DESIGN MODEL**

## DATA DESIGN

USER INFO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR**  **OPTIONAL(O)** | **PRIMARY**  **KEY** |
| 1.  2.  3.  4. | USER  NAME  USER  ID  PASSWORD  USERTYPE | CHAR  NUMERIC  NUMERIC  CHAR | 30  4  6  2 | M  M  M  M | --  YES  --  -- |

COURSE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR**  **OPTIONAL(O)** | **PRIMARY**  **KEY** |
| 1.  2. | COURSE  CODE    COURSE  NAME | NUMERIC  CHAR | 2  30 | M  O | YES  --- |

SEMESTER

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR OPTIONAL(O)** | **PRIMARY**  **KEY** |
| 1. | SEM NO. | NUMERIC | 1 | M | YES |

SUBJECTS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR**  **OPTIONAL(O)** | **PRIMARY**  **KEY** |
| 1.  2. | SUBJECT  NAME  SUBJECT  CODE | CHAR  NUMERIC | 30  1 | O  M | ----  YES |

FACULTY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR**  **OPTIONAL(O)** | **PRIMARY**  **KEY** |
| 1.  2. | FACULTY  NAME  FACULTY  CODE | CHAR  NUMERIC | 30  4 | O  M | ---  YES |

FACULTY & SUBJECTS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR**  **OPTIONAL(O)** | **PRIMARY**  **KEY** |
| 1.  2. | FACULTY  CODE  SUBJECT CODE | NUMERIC  NUMERIC | 4  3 | M  M | YES  YES |

STUDENT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR**  **OPTIONAL(O)** | **PRIMARY**  **KEY** |
| 1.  2.  3.  4.  5.  6.  7.  8. | NAME  FATHER  NAME  MOTHER  NAME  ROLL NO.  UNIV NO.  ADDRESS  PHONE NO.  BIRTHDATE | CHAR  CHAR  CHAR  NUMERIC  NUMERIC  ALPHANUMERIC  NUMERIC  DATE | 30  30  30  5  6  30  8 | M  O  O  M  M  O  O  O | ---  ---  ---  YES  ---  ---  ---  --- |

STUDENT’S ATTENDANCE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR**  **OPTIONAL(O)** | **PRIMARY**  **KET** |
| 1.  2.  3.  4. | SUBJECT CODE  ROLL NO.  TOTAL LECTURES  LECTURES  ATTENDED | NUMERIC  NUMERIC  NUMERIC  NUMERIC | 2  3  4  2 | M  M  O  O | YES  YES  --  -- |

STUDENT’S ASSIGNMENTS / PROJECT MARKS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR**  **OPTIONAL(O)** | **PRIMARY**  **KEY** |
| 1.  2.  3.  4. | SUBJECT CODE  ROLL NO.  TOTAL  MRKS  MARKS  SCORED | NUMERIC  NUMERIC  NUMERIC  NUMERIC | 2  3  4  2 | M  M  M  M | YES  YES  YES  -- |

STUDENT’S CLASS TEST MARKS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR**  **OPTIONAL(O)** | **PRIMARY**  **KEY** |
| 1.  2.  3.  4. | SUBJECT CODE  ROLL NO.  MAX MARKS  MARKS SCORED | NUMERIC  NUMERIC  NUMERIC  NUMERIC | 2  3  4  2 | M  M  M  M | YES  YES  --  -- |

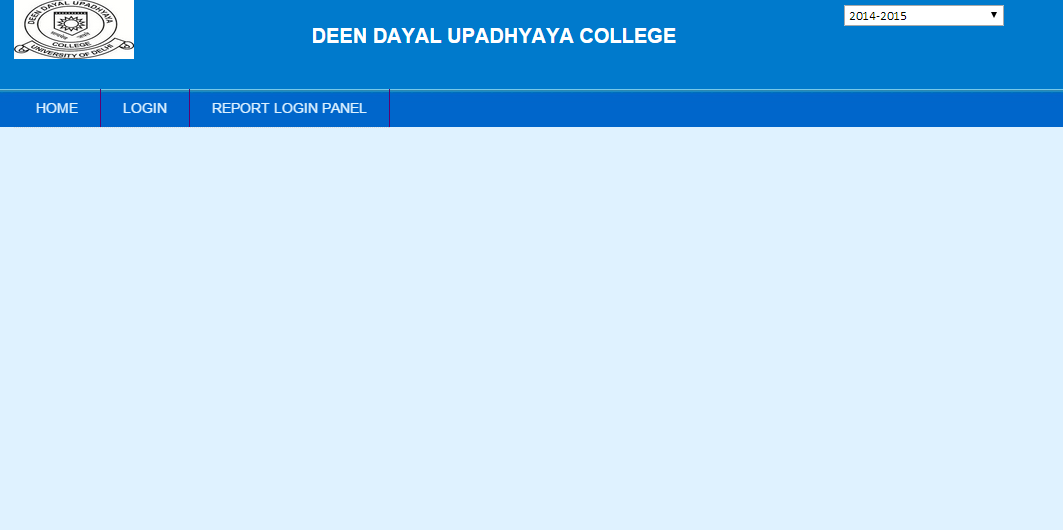
STUDENT INTERNAL ASSESSMENT RECORD

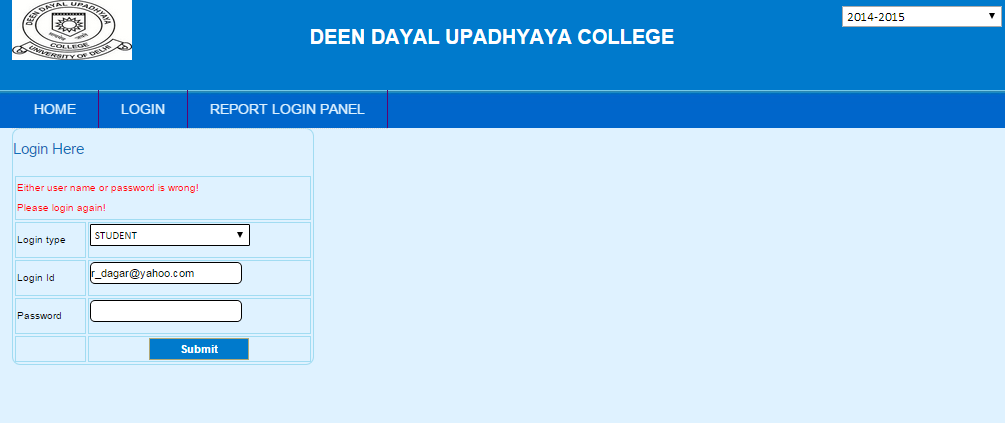
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S NO.** | **FIELD** | **TYPE** | **LENGTH** | **MANDATORY(M)**  **OR**  **OPTIONAL(O)** | **PRIMARY**  **KEY** |
| 1.  2.  3.  4.  5.  6. | SUBJECT CODE  ROLL NO.  ASSIGNMENT MARKS  (OUT OF 10)  CLASS TEST MARKS  (OUT OF 10)  ATTENDANCE MARKS  (OUT OF 5)  TOTAL MARKS  (OUT OF 25) | NUMERIC  NUMERIC  NUMERIC  NUMERIC  NUMERIC  NUMERIC | 3  4  2  2  1  2 | M  M  M  M  M  M | YES  YES  --  --  --  -- |

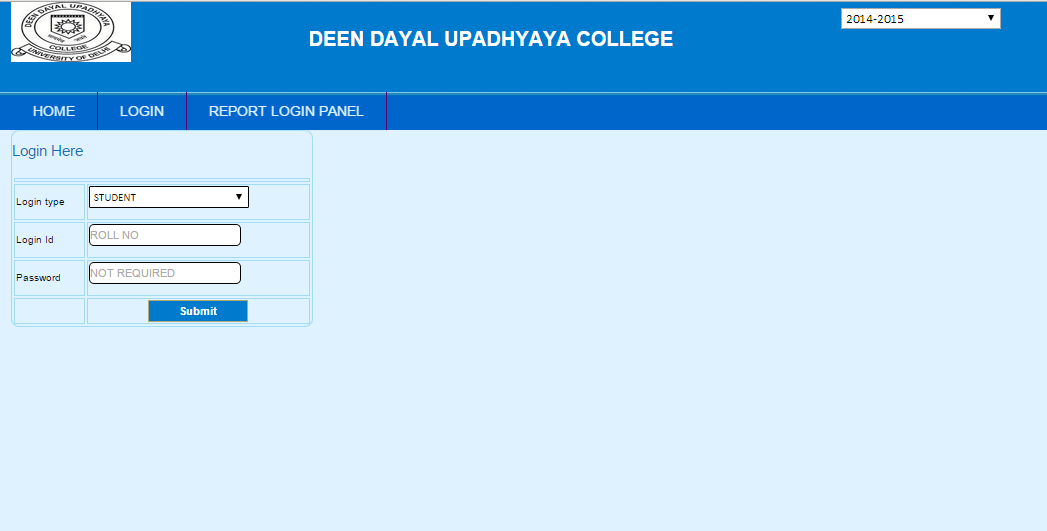
* 1. **ARCHITECTURAL DESIGN**

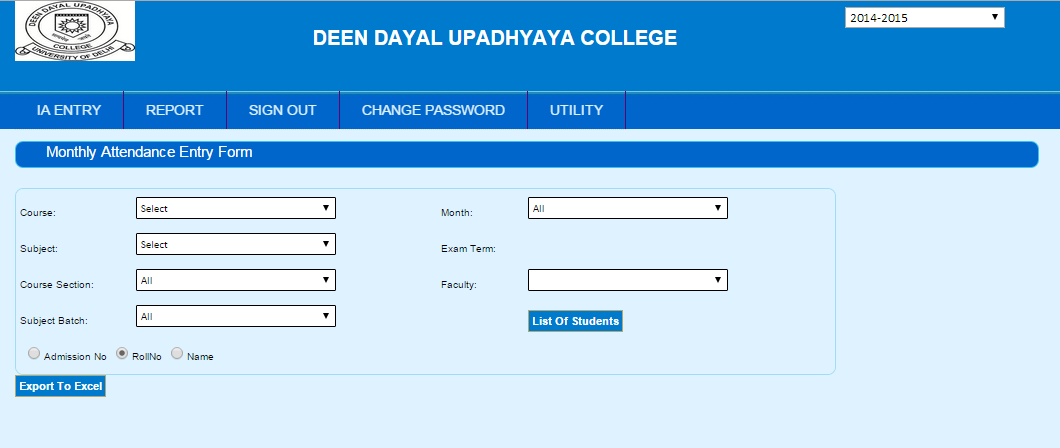


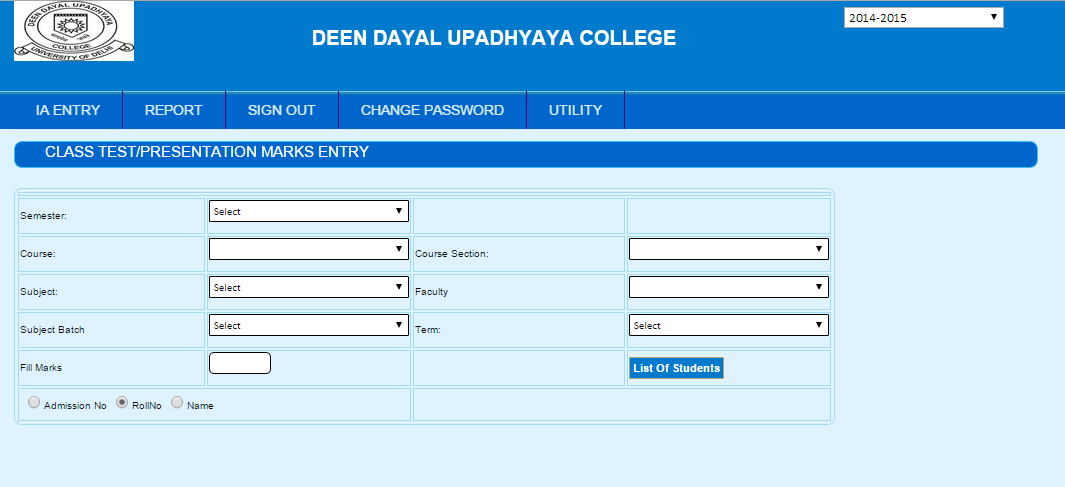
* 1. **INTERFACE DESIGN**

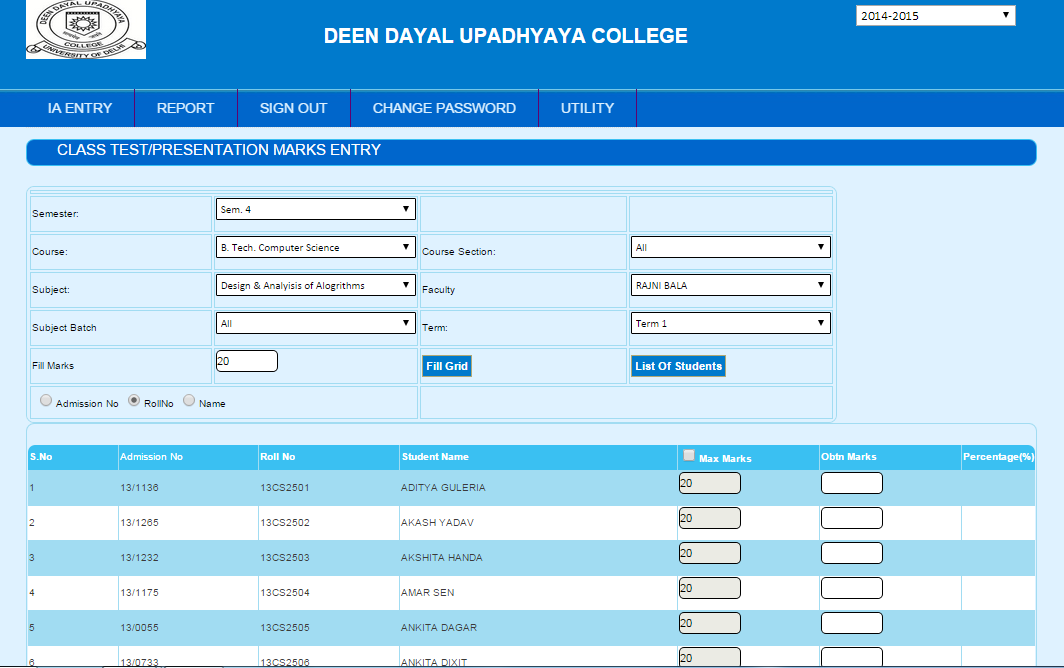












* 1. **PSEUDO CODE FOR FACULTY**

1. Faculty login using username and password
2. If invalid(username, password) “login unsuccessful”
3. Else login successful
4. Faculty screen displayed
5. If Faculty selects IA Entry
   1. If Faculty selects Monthly Attendance Entry
      1. Monthly attendance form displayed
      2. User selects course, semester, subject , month and clicks on list of students
      3. List of students displayed.
      4. User fills the attendance and clicks on submit button.
      5. If in any record (total\_attendance< classes\_attended) display error
      6. Else records saved successful
   2. Else if faculty selects Class-test marks
      1. Class-test form displayed
      2. User selects course, semester, subject and clicks on list of students
      3. List of students displayed.
      4. User fills the maximum marks, marks obtained and clicks on submit button.
      5. If in any record (maximum\_marks< marks\_obtained) display error
      6. Else records saved successful
   3. Else if faculty selects Assignment/project marks
      1. Assignment form displayed
      2. User selects course, semester, subject and clicks on list of students
      3. List of students displayed.
      4. User fills the maximum marks, marks obtained and clicks on submit button.
      5. If in any record (maximum\_marks< marks\_obtained) display error
      6. Else records saved successful
6. If Faculty selects Report
   1. List of available report displayed
   2. User selects desired report
   3. Report is displayed on screen
7. If faculty selects change password
   1. Faculty is asked to enter old password and new password
   2. Faculty clicks on change
   3. Password is changed.
8. If faculty clicks on signout he/she is logged out.

**CHAPTER 5**

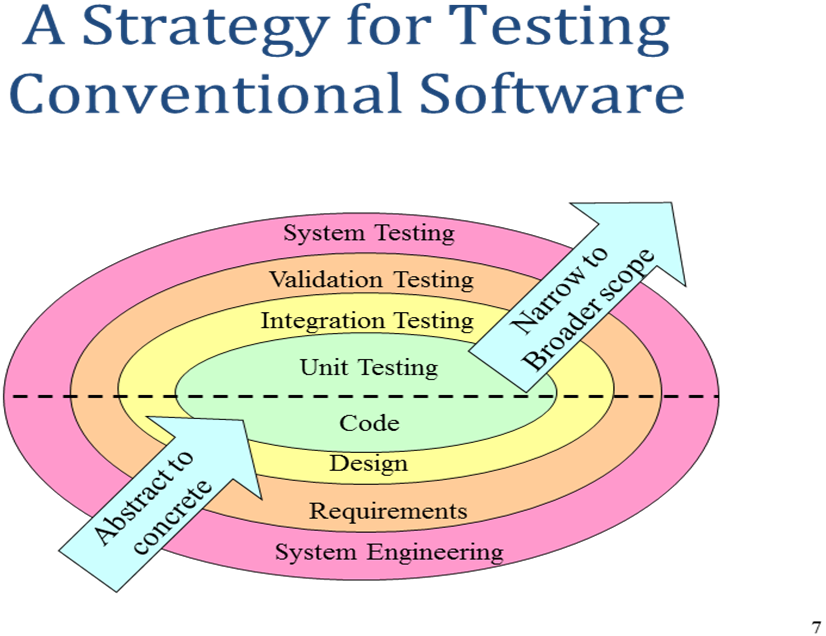
**SOFTWARE**

**TESTING**

Software testing is basically the process of executing program with the intention of finding errors in the code. It is required before launching of software project. Key features of testing are as follows:

* To perform effective testing, a software team should conduct effective formal technical reviews.
* Testing begins at the component level and work outward toward the integration of the entire computer-based system.
* Different testing techniques are appropriate at different points in time.
* Testing is conducted by the developer of the software and (for large projects) by an independent test group (ITG).
* Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.

**CONVENTIONAL STRATEGY FOR TESTING**

****

## SOFTWARE TESTING STRATEGIES

**Unit testing**

* Concentrates on each component/function of the software as implemented in the source code

**Integration testing**

* Focuses on the design and construction of the software architecture

**Validation testing**

* Requirements are validated against the constructed software

**System testing**

* The software and other system elements are tested as a whole

## SOFTWARE TESTING TECHNIQUES:

**Black-box testing**

* Knowing the specified function that a product has been designed to perform, test to see if that function is fully operational and error free
* Includes tests that are conducted at the software interface
* Not concerned with internal logical structure of the software

**White-box testing**

* Knowing the internal workings of a product, test that all internal operations are performed according to specifications and all internal components have been exercised
* Involves tests that concentrate on close examination of procedural detail
* Logical paths through the software are tested
* Test cases exercise specific sets of conditions and loops

## SAMPLE UNIT TEST CASE

|  |  |  |
| --- | --- | --- |
| **S.No** | **Test Case** | **Expected Result** |
| 1. | Enter valid username and password | Login successful. |
| 2. | Enter valid username and invalid password | Error message: wrong password. |
| 3. | Login as admin  Insert a new course | New record inserted |
| 4. | Login as admin  Insert a new student | New record inserted |
| 5. | Login as admin  Insert a new subject | New subject inserted |
| 6. | Login as admin  Insert a new faculty | New faculty inserted |
| 7. | Login as faculty  Click on Attendance  Select the course, sem ,paper, month | List of students in specified course |
| 8. | Login as faculty  Click on Attendance  Select the course, sem ,paper, month  Fill the attendance  Click on save | Attendance of the students should be saved in database. |
| 9. | Login as faculty  Click on Attendance  Select the course, sem ,paper, month for which attendance has already been submitted | List of students with their attendance. |
| 10. | Login as faculty  Click on class-test marks  Select the course, sem ,paper  Fill the maximum marks, marks obtained  Click on save | Class-test marks of students should be saved. |
| 11. | Login as Student  Click on view attendance  Select month | Attendance of student in all subject in the specified month should be displayed |
| **S.No** | **Test Case** | **Expected Result** |
| 12. | Login as Student  Click on view internal assessment marks | Internal assessment in all the subjects should be displayed. |
| 13. | Allow 4-5 users to login simultaneously. | Login should be successful. |
| 14. | Login as Faculty  Click on Attendance  Modify some attendance  Click on save | Modification successful. |
| 15. | Login as faculty  Click on class-test  Enter marks obtained greater that total marks | Error: marks obtained can’t be greater than total marks. |

## CYCLOMATIC COMPLEXITY

Enter Username and Password

**1**

**FLOW CHART**

Region 7

**18**

**21**

**20**

**17**

**16**

**15**

**14**

**13**

**12**

**11**

**10**

**9**

**8**

**7**

**6**

**5**

**19**

**4**

**3**

**2**

Region 5

Region 6

Err

Err

Err

No

No

No

Region 4

Success

Signout

C

B

A

Exit

Submit

Submit

Submit

Test

Assignment

Attendance

Enter data

Enter data

Enter data

IA

Reports

IA/

Reports/

Pswd change/

Signout

Attend./

Test/

Assign.

Region 3

Region 2

Region 1

Password change

Enter new Pswd & old Pswd

No

Signout

Display Reports

Valid

Yes

**FLOW GRAPH**

Region 3

Region 2

Region 7

Region 6

Region 5

Region 4

Region 1

There are 3 methods to calculate **cyclomatic complexity**:

* By calculating number of regions = 7

**Cyclomatic complexity = 7**

* By calculating number of predicate nodes (i.e. P=6 )

**Cyclomatic complexity** = P+1 = 6+1 = 7

* By calculating number of nodes (N=17) and number of edges (E=22),

**Cyclomatic complexity = E – N + 2 = 22 – 17 + 2 = 7**

**BIBLIOGRAPHY**

* Roger S. Pressman, *“Software Engineering A Practitioner Approach”*
* Pankaj Jalote, *“A concise Introduction to software engineering”*