



**Islamic Emirate of Afghanistan**

**Ministry of Higher Education**

**Ghalib University**

**Faculty of Computer Science**

**Software Engineering Department**

**Final Year Project Report**

**On**

**Academy Management System**

**For**

**BCS (Bachelor of Computer Science)**

**Submitted by:** Sirajuddin Siraj

**Guided by:** Lec. Zabihullah Rahmani

**Submitted to:** Software Engineering department

**Ghalib University**

**2024, Heart – Afghanistan**

**Autumn 2024**



Islamic Emirate of Afghanistan

Ministry of Higher Education

Ghalib University

Faculty of Computer Science

Software Engineering



## Confirmation of Jury members at thesis defense for Bachelor Degree

The jury members approve the final version of **Mr. Sirajuddin Siraj S/O Ala Uddin** the fulfillment of bachelor degree in Computer Science, Software Engineering Department titled

(AMS)

“Academy Management System”

(سیستم مدیریتی اکادمی)

Name, Last Name

Academic Grade

Signature

Supervisor: .....

.....

.....

Member: .....

.....

.....

Member: .....

.....

.....

Head of Department's Approval:

Name

Last Name

Academic Grade

.....

.....

.....

Register No .....

Signature.....

Faculty of Dean's Approval:

Name

Last Name

Academic Grade

.....

.....

.....

Register No .....

Signature.....

# Acknowledgment

---

First of all, I do hereby thank Almighty Allah who has given me health and wellness throughout my life and has helped me a lot in fulfilling my wishes. Allah, who gave me the talent and ability to learn and deal in variety of complex topics during the last four years and enabled me to provide services for my community by preparation of this project,

I express my sincere gratitude and thank to my supervisor Zabihullah Rahmani and all my friends who supported me and encouraged me for my hard work to complete this book. Without their assistance and dedicated involvement in every step throughout the process, this paper would have never been accomplished. I would like to thank you very much for your support and understanding over these past four years.

I would also thank all my family members specially my beloved parents, who have been giving me support and motivation throughout of my project.

# Abstract

---

An academy management system is a software solution designed to help academies to manage their workforce more effectively. The system typically includes a range of tools and features aimed at admission processes, from recruitment new teachers and new Students and onboarding to performance management.

One of the key benefits of an academy management system is that it can help academies to automate their admission processes, reducing the administrative burden on office workers and freeing up their time to focus on more strategic tasks.

Academy management systems may also include features to help academies manage students, classes, teachers benefits and payroll. For example, the system may provide tools for tracking teacher and student attendance and leave, calculating overtime and paid time off of teachers, and automating the payroll process of teachers and office workers. This can help to ensure that teachers are compensated fairly and accurately, while reducing the risk of errors or discrepancies in payroll processing.

Overall, an Academy management system can help academies to manage their workforce more efficiently and effectively, by providing a centralized platform for HR processes and automating many of the routine tasks that can consume HR teams' time and resources. By implementing an academy management system, academies can improve their HR processes, enhance academy management and productivity, and ultimately support their overall educational goals.

## Table of Contents

1	Introduction to System .....	8
1.1	Introduction .....	9
1.2	Problems in Existing System .....	9
1.3	Scope of Proposed System.....	10
1.3.1	Technical Feasibility .....	10
1.3.2	Operating Environment-Hardware and Software.....	10
1.4	Main Objectives .....	10
2	Application Requirements .....	11
2.1	Business Requirements.....	12
2.2	Functional Requirements.....	12
2.3	Non-function Requirements .....	13
2.4	System Analysis and Design .....	13
2.4.1	Data Flow Diagram.....	14
2.4.2	DFD level-0.....	15
2.4.3	DFD level-1 .....	15
2.5	ERD (Entity Relationship Diagram).....	16
2.5.1	Entity .....	16
2.5.2	Attribute.....	16
2.5.3	Entity-Set and Keys.....	17
2.5.4	Relationships .....	17
2.6	Database ER Diagram.....	19
2.7	Database Schema.....	20
2.8	Database Tables Creation .....	20
2.8.1	Students Table .....	20
2.8.2	Course .....	21
2.8.3	Teacher.....	21
2.8.4	Timetable .....	22
2.8.5	TeacherCourse .....	22
2.8.6	Username.....	22
2.8.7	Employee.....	23
3	Test Procedure and Implementation.....	24

3.1	What are Software Testing Procedures? .....	25
3.2	Test Procedure/Script Specification.....	25
3.3	Format for Test Procedure / Script Specification.....	25
3.3.1	Identifier.....	26
3.3.2	Purpose .....	26
3.3.3	Special Requirements.....	26
3.4	Testing Levels .....	26
3.4.1	Unit Testing .....	27
3.4.2	Integration Testing.....	27
3.4.3	Database Testing.....	27
3.4.4	Recovery Testing .....	28
3.4.5	Functional Testing .....	28
3.4.6	Compatibility Testing .....	30
3.4.7	Load Testing .....	31
3.4.8	System Testing .....	31
3.4.9	Performance Testing.....	31
3.4.10	White-Box Testing .....	32
3.4.11	Black-Box Testing .....	32
4	User Interface .....	34
4.1	Login Form .....	35
4.2	Side bar .....	36
4.3	Dashboard.....	37
4.4	Students .....	37
4.5	Course .....	39
4.6	Time Table.....	40
4.7	Teacher .....	41
4.8	TeacherCourse .....	42
4.8.1	Employee.....	43
5	Conclusion and Future Work .....	44
5.1	Conclusion.....	44
5.2	Future Work .....	44

## Table of Figures

Chapter 2. Figure 2-1 DFD level-0.....	15
Chapter 2. Figure 2-3 ER Diagram.....	19
Chapter 2. Figure 2-4 Database Design.....	20
Chapter 4. Figure 4-1 Log in form .....	35
Chapter 4. Figure 4-2 Side bar .....	36
Chapter 4. Figure 4-3 Dashboard page .....	37
Chapter 4. Figure 4-4 Students .....	38
Chapter 4. Figure 4-5 courses Page .....	39
Chapter 4. Figure 4-6 time table .....	40
Chapter 4. Figure 4-7 Teacher.....	41
Chapter 4. Figure 4-8 TeacherCourse .....	42
Chapter 4. Figure 4-9 Employee List .....	43

## Table of Tables

Table 2-1 Student.....	21
Table 2-2 Course .....	21
Table 2-3 Teacher.....	22
Table 2-4 Time table .....	22
Table 2-5 TeacherCourse .....	22
Table 2-6 username .....	23
Table 2-7 Employee.....	23

*Chapter-1*

# **1 Introduction to System**



## 1.1 Introduction

Effective management of an academy's workforce is essential to achieve optimal performance and growth. Academy management systems play a crucial role in managing various aspects of students, teacher and employees, including payroll, benefits, attendance, performance, and more. In today's fast-paced and dynamic educational environment, traditional manual methods of managing students and teachers' data have become inefficient and time-consuming, leading to increased administrative burdens and reduced productivity.

The purpose of this proposal is to introduce an Academy Management System (AMS) that can help academies streamline and automate their students and teachers-related processes. This system can provide significant benefits to academies of all sizes, from small academies to large academies.

The proposed Academy Management System will enable academies to manage their workforce more efficiently and effectively, allowing them to focus on their core educational activities. The system will provide real-time access to students and teachers data, enabling HR managers to make informed decisions, improve students and teachers' satisfaction, and enhance overall productivity.

The proposed system will include several features, such as a student, teacher, courses and office workers database, attendance tracking, performance management, payroll processing, and benefits administration. The system will be user-friendly and accessible from any device that have windows OS, ensuring ease of use and convenience for all users.

In conclusion, the proposed Academy Management System will provide a comprehensive and efficient solution for managing student, teacher, employees and courses' data, enabling academies to streamline their HR processes and improve overall performance. The system's features and benefits make it an ideal solution for Academies seeking to optimize their workforce management and stay ahead of the competition.

## 1.2 Problems in Existing System

Similar systems had been built before with the mentioned functionalities, although in different technologies and languages. This system will be built using various technologies. These technologies will bring about superb performance to the system. In addition, our society is not used to using modern technologies in its business. One of the keys to the

growth of companies, educational institutes and organizations is to automate the management of employees, students, etc. HR management. Consequently, this system will introduce a contemporary way of managing for society and business owners.

## 1.3 Scope of Proposed System

As the problems mentioned in the above section, here I will elaborate on all the features and functionalities that ease the processes in all dimensions of the Academy. I will elaborate Student/Teacher Management, Course management, Employee Management, user-friendly interface and system security.

When new Student/Teacher is added to the list, user will be able add Course to a specific Teacher, view his/her Courses and print the report on student. The report can be printed by searching student using his/her ID, Name, Semester as well.

### 1.3.1 Technical Feasibility

The old flat file system is substituted by our application and required some hardware and software infrastructures and human resources. Here I will explain all the requirements that are necessary for our software.

Our application requires 1 trained staff to enter data and run the system perfectly. The staff handles the administrative activities.

### 1.3.2 Operating Environment-Hardware and Software

- Hardware requirements for Server
  - Processor: Core i3 or more for optimum performance
  - Ram: 2GB| 4GB is recommended
  - Hard Disk: at least 2GB space
- Software requirements for server
  - any operating system
  - Support and run .net Applications and SQL server database
  - Provide services to client requests

## 1.4 Main Objectives

- Reduce time consumption on flat file system
- Reduce complexity on generating reports
- Reduce ambiguity and doubt among partners with clear understanding of company status
- Enhance decision making process
- Enhance coordination between customers and rooms
- Easy and accurate employee and cash management

*Chapter-2*

## **2 Application Requirements**

The first and fundamental steps to develop a system are gathering the requirements from different people in the organization and analyze them from a variety of dimensions. In this chapter, we focus and elaborate on the function, non-function and business requirements in detail.

## 2.1 Business Requirements

- Develop a system that eliminates complexity
- Develop a centralized system that can be accessed from anywhere
- Better and accurate cash and safe management
- Develop a system that is easy to make decisions
- Develop a system that is easy to use

## 2.2 Functional Requirements

- i. Login form of the System
  - Admin can login to the System and this system has various admin to manage their employee sections.
- ii. Language preferences
  - It only supports English language
- iii. Employee Management
  - Add new employee
  - Delete or update employees
  - Search employee in multiple ways like by (name, father's name, status)
- iv. Custom Page titles
  - Each Page has a custom name.
- v. Admin Panel
  - Admin can Search Employees and Teachers and Students
- vi. Reports
  - Teacher Report
  - Student Report
  - Courses given to Teachers Report

## 2.3 Non-function Requirements

### vii. Security

- Security is the most important feature in the system. Although the security of a system is vulnerable, I try my best to make the application secure by different techniques like using encryption on momentous data, preventing the SQL Injection, restricting the user's privileges, using Identity API for user sign up/sign in process and much more...

### viii. Application Accessibility

- End-users want easiness and friendly systems that he/she can access the application through different operating systems and different devices with their sizes. So, the system that I am going to build is accessible via browser and it's compatible with different OS and fully responsive in different sizes.

### ix. Guarantee

- This system will be developed by standard technology and accepted methods used in different countries. Your specialists are free to check the quality of the application after deployment. So any problem or error will be fixed free of cost but any enhancement requires further consultation.

### x. Software Process Model

- There are different process models but I really like waterfall modeling. By gathering the requirements and analyzing them, I will present in the department. If the presentation of engineering processes and models are accepted then the development cycle will be started by me.

### xi. Technologies

- Every day new models, processes and programming languages are developing. So being up to date is my commitment. For frontend, I used C# in ASP .net. Also, for storing data I used Microsoft SQL server.

## 2.4 System Analysis and Design

Software design is the process by which an agent creates a specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints. Software design may refer to either "all the activities involved in

conceptualization, framing, implementing, commissioning and ultimately modifying complex systems".

Software design usually involves problem-solving and planning a software solution. This includes both a low-level component and algorithm design and a high-level, architecture design.

### **2.4.1 Data Flow Diagram**

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

#### **2.4.1.1 DFD rules and tips**

- Each process should have at least one input and an output.
- Each data store should have at least one data flow in and one data flow out.
- Data stored in a system must go through a process.
- All processes in a DFD go to another process or a data store.

#### **2.4.1.2 DFD Symbols and Notations**

- Squares
- Rounded Rectangle
- Arrows
- Open-Ended rectangles

### 2.4.2 DFD level-0

This DFD level focuses on high-level system processes or functions and the data sources that flow to or from them. Level 0 diagrams are designed to be simple, straightforward overviews of a process or system.

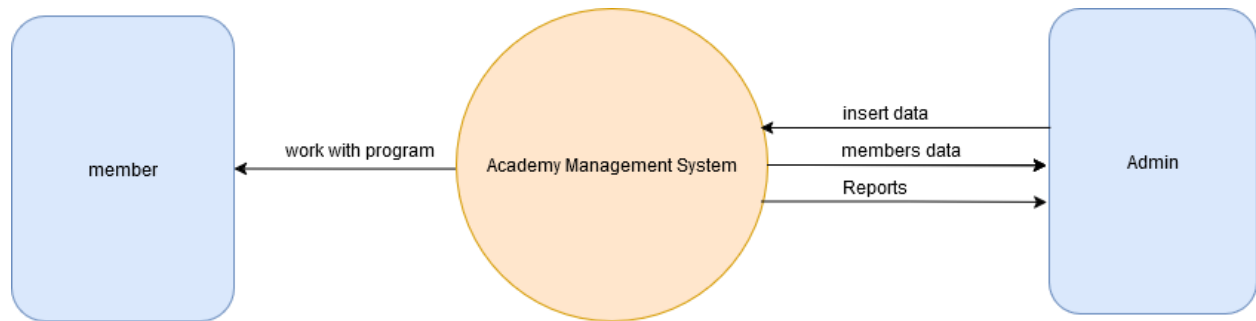


Figure 2-1 DFD level-0

### 2.4.3 DFD level-1

While level 1 DFDs are still broad overviews of a system or process, they're also more detailed — they break down the system's single process node into subprocesses.

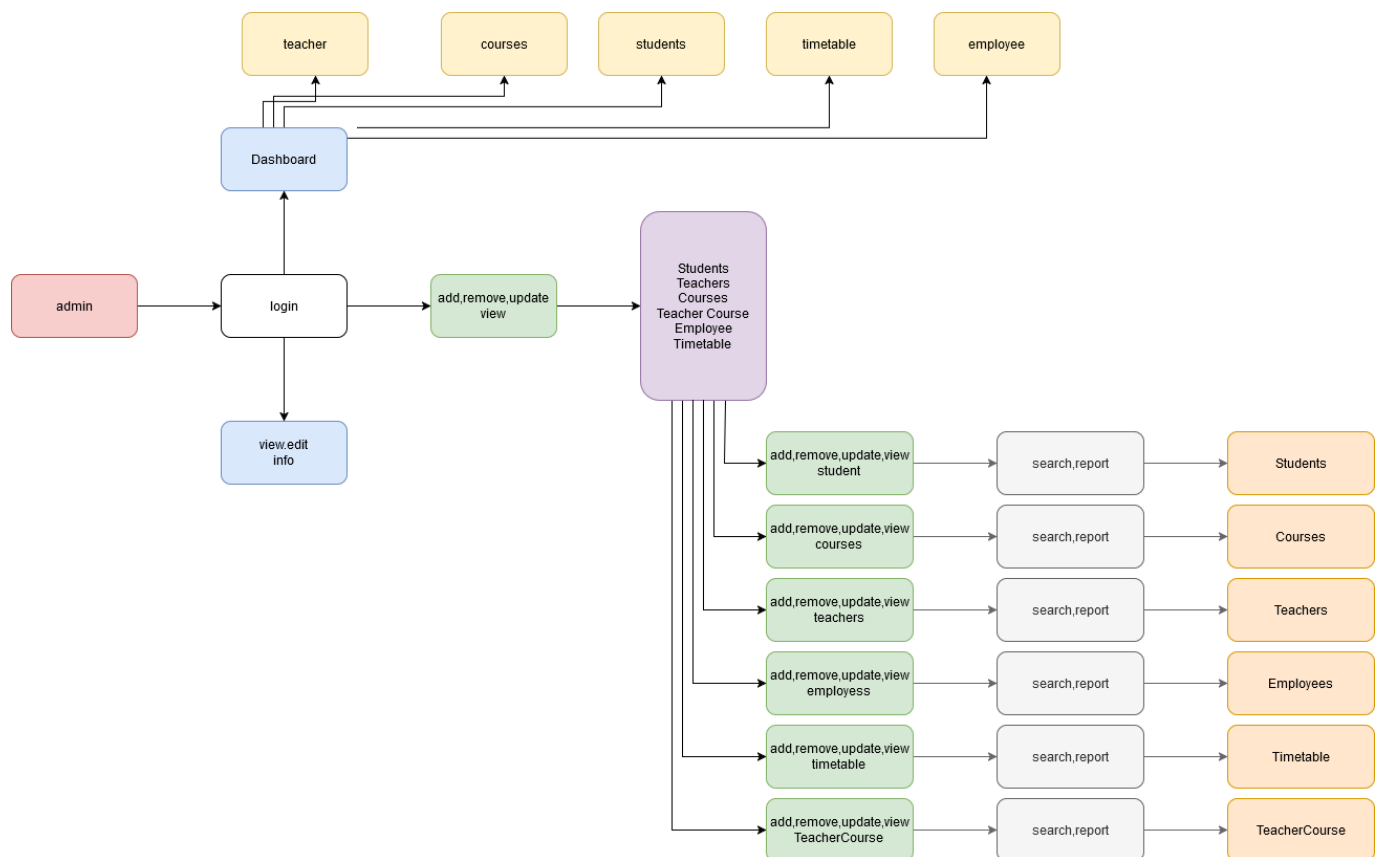


Figure 2-2 DFD level-1

## 2.5 ERD (Entity Relationship Diagram)

ERD is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize the bellow information:

- Entities with system scope
- Inter-relationships among those entities

### 2.5.1 Entity

An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.

An entity set is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values. For example, a student's set may contain all the students of a school; likewise, a teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

### 2.5.2 Attribute

Also known as a column, an attribute is a property or characteristic of an entity. An attribute has a name that describes the property and a type that describes the kind of attributes such as varchar for storing string, int for integer and etc...

When an ERD is drawn for physical database development, it's important to ensure that we use appropriate and supported type of attributes for target RDBMS.

#### 2.5.2.1 Types of Attributes

- **Simple attribute** – Simple attributes are atomic values, which cannot be divided further. For example, a customer's phone number is an atomic value of 10 digits.
- **Composite attribute** – Composite attributes are made of more than one simple attribute. For example, a customer's complete name may have first\_name and last\_name.



- **Derived attribute** – Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average\_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data\_of\_birth.
- **Single-value attribute** – Single-value attributes contain single value. For example – Social\_Security\_Number.
- **Multi-value attribute** – multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email\_address, etc.

### 2.5.3 Entity-Set and Keys

Key is an attribute or collection of attributes that uniquely identifies an entity among entity set. For example, the ID of a customer makes him/her identifiable among customers.

- **Primary Key** – Also known as PK, a primary key is special kind of entity's attribute that uniquely defines a record in a database table. In other words, there must not be two or more records that share same value for primary key attribute.
- **Foreign Key** – Also known as FK, a foreign key is a reference to a primary key in a table. It is used to identify the relationships between entities. Note that foreign key is not unique value and multiple records can share same values.

### 2.5.4 Relationships

A relationship between two entities signifies that two or more entities are associated with each other somehow.

#### 2.5.4.1 Types of Relationship

- Unary or recursive relationship
- Binary relationship
- Ternary relationship
- N-Array relationship

##### 2.5.4.1.1 Unary Relationship

A relationship between the instances of only one entity type is called unary or recursive relationship.

Note: only one entity participates in unary relationship

##### 2.5.4.1.2 Binary Relationship

A relationship between the instances of two entity types is called binary relationship. Or in other words if two tables in database participate in relationship it's called binary relationship.

##### 2.5.4.1.3 Ternary Relationship

A relationship between the instances of three entity types is called binary relationship.

#### 2.5.4.1.4 N-Array Relationship

Participating more than three entity types in a relationship is called N-Array relationship. Note that this kind of relationship is not common because most of relationships in ERD are binary or ternary.

#### 2.5.4.2 Types of Relationship Cardinality

- One to one
- One to many
- Many to Many

##### 2.5.4.2.1 One to Many

In this relationship one instance of an entity can be link with many instances of another entity.

##### 2.5.4.2.2 Many to Many

In this relationship one instance of an entity can be link with many instances of another entity and vice versa.

This developed system contains the bellow entities and its relationships shown as a figure

## 2.6 Database ER Diagram

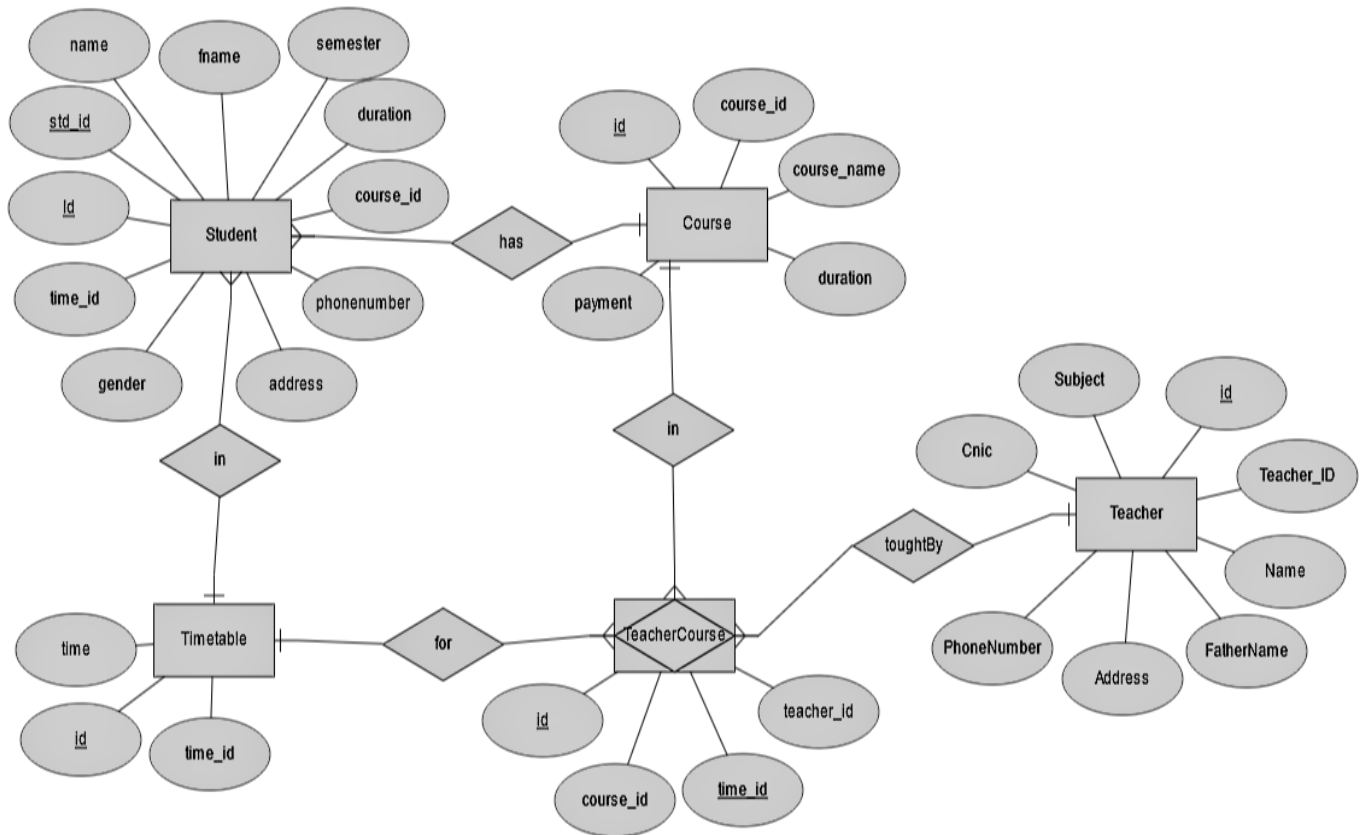


Figure 2-3 ER Diagram

## 2.7 Database Schema

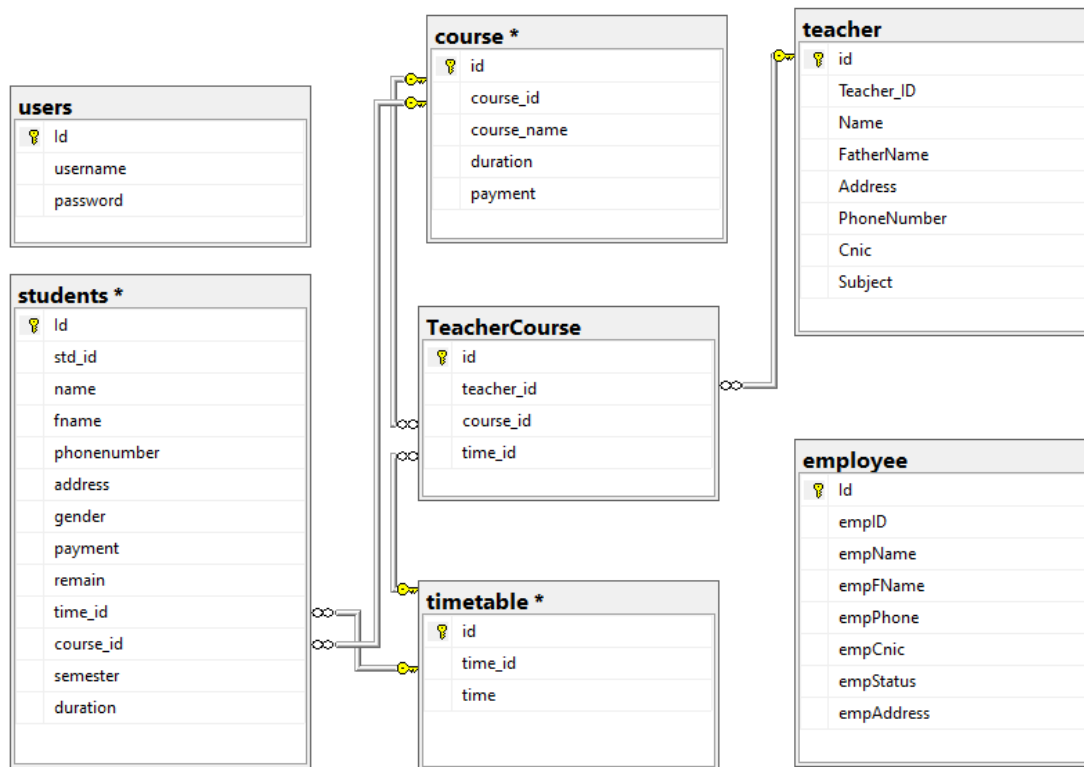


Figure 2-4 Database Design

## 2.8 Database Tables Creation

As I attached Entity relationship diagram and ER Model now in this section, I will elaborate all the tables' attributes and usage in details.

### 2.8.1 Students Table

In this table we add all the Students with full details. For Example, Name, address, contact number etc. which are important for both report printing and other purpose.

Column	Type	Null	Default
Id	Int	No	
Std_id	Int	No	
Name	varchar(50)	No	

Fname	varchar(50)	No	
Phonenumber	Int	No	
Address	varchar(MAX)	No	
Gender	varchar(10)	No	
Payment	Varchar(20)	No	0
Remain	Int	No	
Time_id	Int	No	
Course_id	Int	No	
Semester	Int	NO	
Duration	Varchar(10)	No	

Table 2-1 Student

### 2.8.2 Course

In this table all Courses are recorded with details which will be assign to Teacher.

Column	Type	Null	Default
Id	Int(10)	No	
course_id	Int	No	
course_name	varchar(MAX)	No	
duration	varchar(50)	No	
payment	Int	No	

Table 2-2 Course

### 2.8.3 Teacher

In this table the Teachers' data is stored like teacherID, Name and Father's name etc..

Column	Type	Null	Default
Id	Int	No	
Teacher_ID	Int	No	
Name	Varchar(50)	No	

FatherName	Varchar(50)	No	
Address	Varchar(Max)	No	
PhoneNumber	Bigint	No	
Cnic	Varchar(20)	No	
Subject	Varchar(50)	No	

Table 2-3 Teacher

### 2.8.4 Timetable

In this table the order of time is saved that on which time, an academy can have classes.

Column	Type	Null	Default
Id	Int	No	
Time_id	Int	No	
Time	Varchar(50)	No	

Table 2-4 Time table

### 2.8.5 TeacherCourse

This table contains the list of Courses(books) that are given to the teachers. Each teacher can have multiple classes in a day.

Column	Type	Null	Default
Id	Int	No	
Teacher_id	Int	No	
Course_id	Int	No	
Time_id	Int	No	

Table 2-5 TeacherCourse

### 2.8.6 Username

This table contains the list of available admins through which you can enter to the system.

Column	Type	Null	Default
Id	Int	No	
Username	Nvarchar(50)	No	

Password	Nvarchar(50)	No	
----------	--------------	----	--

Table 2-6 username

### 2.8.7 Employee

All the data about the employees are listed here like about their name, address, status, etc.

Column	Type	Null	Default
Id	Int	No	
empID	Int	No	
EmpName	Varchar(50)	No	
empFName	Varchar(50)	No	
EmpPhone	Int	Yes	Null
empCnic	Varchar(50)	No	
EmpStatus	Varchar(50)	No	
empAddress	Varchar(50)	No	

Table 2-7 Employee

*Chapter 3*

## **3 Test Procedure and Implementation**



For the success and effectiveness of any software and application, it is vital for the team to prepare detailed, informative, & quality documents and reports, which can be used as a deliverable and for communicating the process and procedure of testing. The purpose of test procedure/script specification document is the same and it tries to define the various types of testing techniques used by software testers to ensure the quality and efficiency of the software. As one of the reports prepared after the completion of software testing, the significance of test procedure or script specification is immense and is described in detail below.

### **3.1 What are Software Testing Procedures?**

The various testing practices, processes, & techniques used by testers to ensure that the software application is tested and validated before its release or deployment is known as software testing procedure. It is a combination of several test cases based on a certain logical reason. These are complete, self-contained, self-validating and can be executed automatically, with the assistance of automated tools. Moreover, the software testing procedures are the deliverables produced of the software development process and are used for both initial evaluation and subsequent regression testing of target program module or modifications. Hence, software testing procedures must be defined, planned, constructed, tested & reported regularly to achieve desired results.

### **3.2 Test Procedure/Script Specification**

As reporting and documenting software testing procedure is important for the success of the software testing and development process, the team publishes a test procedure/script specification report, which is a document providing detailed instructions for the execution of one or more test cases. This report defines the purpose of various software testing techniques used by testers and makes them understandable for the client and other stakeholders of the project. IEEE standard for Software Test Documentation (829-1998) defines the format for this report, which is used rigorously followed by individuals/testers all over the globe.

### **3.3 Format for Test Procedure / Script Specification**

With its aim to define and specify the steps used for executing test cases as well as the measures taken to analyze the software item in order to evaluate its set of features, the test procedure / script specification document is prepared by software testers after the accomplishment of the software testing phase. The template/format of this document is universally acknowledged and accepted as it is defined by the IEEE standard 829-1998. Therefore, the format for test procedure / script specification is:

### 3.3.1 Identifier

To avoid confusion each test procedure / script specification document is provided a unique company generated number, which helps define procedure specification level as well as the software it is related to.

### 3.3.2 Purpose

Once a distinctive identification is generated for the document, the purpose of test procedure is defined. It consists of a detailed list of all the test cases covered during testing procedure as well as the description of each test procedure.

### 3.3.3 Special Requirements

Any special requirements and specification mentioned by the client or stakeholders of the project before the commencement of the testing process are recorded here with proper evidence and documentation. Hence, the details included here are:

- Type of testing: Manual or automated
- The test environment
- Any prerequisites of the test procedure.
- The stages where the test is to be executed, such as pre-testing, regression testing, future compliance testing, etc.
- It also includes details about the special skills and training required by the team for the test procedure.

## 3.4 Testing Levels

- Unit Testing
- Integration Testing
- Database Testing
- Recovery Testing
- Functional Testing
  - Smoke Test
  - Sanity Test
- Compatibility Testing
- Load Testing
- System Testing
- Performance Testing
- User Acceptance Testing

### 3.4.1 Unit Testing

It is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

### 3.4.2 Integration Testing

Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. Integration Testing focuses on checking data communication amongst these modules. Although each software module is unit tested, defects still exist for various reasons like:

- A Module, in general, is designed by an individual software developer whose understanding and programming logic may differ from other programmers. Integration Testing becomes necessary to verify the software modules work in unity
- At the time of module development, there are wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence system integration Testing becomes necessary.
- Interfaces of the software modules with the database could be erroneous
- External Hardware interfaces, if any, could be erroneous
- Inadequate exception handling could cause issues.

### 3.4.3 Database Testing

The GUI is in most cases given the most emphasis by the respective test managers as well as the development team members since the Graphical User Interface happens to be the most visible part of the application. However, what is also important is to validate the information that can be considered as the heart of the application DATABASE. Let us consider a Banking application whereby a user makes transactions. Now from database testing viewpoint following things are important:

- The application stores the transaction information in the application database and displays them correctly to the user.
- No information is lost in the process.
- No partially performed or aborted operation information is saved by the application.

- No unauthorized individual is allowed to access the user's information.

Database Testing is checking the schema, tables, triggers, etc. of the database under test. It may involve creating complex queries to load/stress test the database and check its responsiveness. Database testing analyzes data integrity and consistency and has 3 types.

1. Structural Testing
2. Functional Testing
3. Non-functional Testing

### 3.4.4 Recovery Testing

There is an array of software testing to ensure its efficiency. One of these testing techniques is non-functional testing. A non-functional testing is a term used for the part of the software that is not connected to a specific user action or function such as security. This is further sub-divided into many testing methods among which is Recovery testing.

Recovery testing is basically done in order to check how fast and better the application can recover against any type of crash or hardware failure etc. Type or extent of recovery is specified in the requirement specifications. It is basically testing how well a system recovers from crashes, hardware failures, or other catastrophic problems. Recovery testing is an effective way to check the recovery rate of an application when it has suffered a huge blow of hardware failure or crash. It is a way of manually stimulating the failure of the software to test the possibility and the rate of its recovery.

#### 3.4.4.1 Steps

- Determining the feasibility of the recovery process.
- Verification of the backup facilities.
- Ensuring proper steps are documented to verify the compatibility of backup facilities.
- Providing Training within the team.
- Demonstrating the ability of the organization to recover from all critical failures.
- Maintaining and updating the recovery plan at regular intervals.

### 3.4.5 Functional Testing

It is a type of software testing which verifies that each function of the software application operates in conformance with the functional requirements. All functionality of the system is tested by providing appropriate input, verifying the output, and comparing the actual results with the expected results. This testing mainly involves black box testing and it is not concerned about the source code of the application. This testing involves checking of User Interface, APIs, Database, security, client/ server applications and

functionality of the Application under Test. The testing can be done either manually or using automation. Typically, functional testing involves the following steps:

- Identify functions that the software is expected to perform.
- Create input data based on the function's specifications.
- Determine the output based on the function's specifications.
- Execute the test case.
- Compare the actual and expected outputs.

### 3.4.5.1 Smoke Testing

It is also known as “Build Verification Testing”, is a type of software testing that comprises of a non-exhaustive set of tests that aim at ensuring that the most important functions work. The result of this testing is used to decide if a build is stable enough to proceed with further testing. The term ‘smoke testing’, it is said, came to software testing from a similar type of hardware testing, in which the device passed the test if it did not catch fire (or smoked) the first time it was turned on.

Smoke testing covers most of the major functions of the software but none of them in depth. The result of this test is used to decide whether to proceed with further testing. If the smoke test passes, go ahead with further testing. If it fails, halt further tests and ask for a new build with the required fixes. If an application is badly broken, detailed testing might be a waste of time and effort.

Smoke test helps in exposing integration and major problems early in the cycle. It can be conducted on both newly created software and enhanced software. Smoke test is performed manually or with the help of automation tools/scripts. If builds are prepared frequently, it is best to automate smoke testing.

#### 3.4.5.1.1 Advantages

- It exposes integration issues.
- It uncovers problems early.
- It provides some level of confidence that changes to the software have not adversely affected major areas (the areas covered by smoke testing, of course)

### 3.4.5.2 Sanity Testing

Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

The objective is "not" to verify thoroughly the new functionality but to determine that the developer has applied some rationality (sanity) while producing the software. For instance, if you're scientific calculator gives the result of  $2 + 2 = 5$ ! Then, there is no point testing the advanced functionalities like  $\sin 30 + \cos 50$ .

For example, in an e-commerce project, main modules are login page, home page, user profile page, user registration etc. There is a defect in the login page when the password field accepts less than four alpha numeric characters and the requirement mentions that this password field should not be below eight characters. Hence, the defect is reported by the testing team to the development team to resolve it. Then the development team fixes the reported defect and sends it to the testing team for clearance. Then the testing team checks whether the changes done are working fine or not. It is also determined if it does have an impact on other related functionalities. Now there is a functionality to update the password in the user profile page. As part of the sanity testing, login page is validated as well as the profile page to ensure that the checks are working fine at both the places.

### 3.4.6 Compatibility Testing

Compatibility Testing is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices. Compatibility Testing is a type of Non-functional testing.

Let's look into compatibility testing types

- **Hardware:** It checks software to be compatible with different hardware configurations.
- **Operating Systems:** It checks your software to be compatible with different Operating Systems like Windows, UNIX, and Mac OS etc.
- **Software:** It checks your developed software to be compatible with other software. For example, MS Word application should be compatible with other software like MS Outlook, MS Excel, and VBA etc.
- **Network:** Evaluation of performance of a system in a network with varying parameters such as Bandwidth, Operating speed, Capacity. It also checks application in different networks with all parameters mentioned earlier.

- **Browser:** It checks the compatibility of your website with different browsers like Firefox, Google Chrome, and Internet Explorer etc.
- **Devices:** It checks compatibility of your software with different devices like USB port Devices, Printers and Scanners, Other media devices and Bluetooth.
- **Mobile:** Checking your software is compatible with mobile platforms like android, iOS etc.
- **Versions of the software:** It is verifying your software application to be compatible with different versions of the software. For instance, checking your Microsoft Word to be compatible with Windows 7, Windows 7 SP1, Windows 7 SP2, and Windows 7 SP3.

### 3.4.7 Load Testing

Load testing is a type of non-functional testing. A load test is type of software testing which is conducted to understand the behavior of the application under a specific expected load. Load testing is performed to determine a system's behavior under both normal and at peak conditions.

Load testing one among the different kinds of performance testing that determines the performance of the system in real time load conditions. It is basically used to ensure that the application performs satisfactorily when many users try to access or use it at the same time.

### 3.4.8 System Testing

System Testing is the testing of a complete and fully integrated software product. Usually, software is only one element of a larger computer-based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing.

### 3.4.9 Performance Testing

Performance Testing is defined as a type of software testing to ensure software applications will perform well under their expected workload. Features and Functionality supported by a software system is not the only concern. A software application's performance like its response time, reliability, resource usage and scalability do matter. The goal of Performance Testing is not to find bugs but to eliminate performance bottlenecks. The focus of Performance Testing is checking a software program

- **Speed -** Determines whether the application responds quickly

- Scalability - Determines maximum user load the software application can handle.
- Stability - Determines if the application is stable under varying loads

### 3.4.10 White-Box Testing

White Box Testing is defined as the testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box tests. It is usually performed by developers.

It is one of two parts of the "Box Testing" approach to software testing. Its counterpart, Black-box testing, involves testing from an external or end-user type perspective. On the other hand, White-box testing is based on the inner workings of an application and revolves around internal testing. White-box test design techniques include the following code coverage criteria:

- Control flow testing
- Data flow testing
- Branch testing
- Statement coverage
- Decision coverage
- Modified condition/decision coverage
- Prime path testing
- Path testing

### 3.4.11 Black-Box Testing

BLACK BOX TESTING, also known as Behavioral Testing is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors



#### 3.4.11.1 Definition by ISTQB

- **Black box testing:** Testing, either functional or non-functional, without reference to the internal structure of the component or system.

#### 3.4.11.2 Advantages

- Tests are done from a user's point of view and will help in exposing discrepancies in the specifications.
- Tester need not know programming languages or how the software has been implemented.
- Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer-bias.
- Test cases can be designed as soon as the specifications are complete.

#### 3.4.11.3 Disadvantages

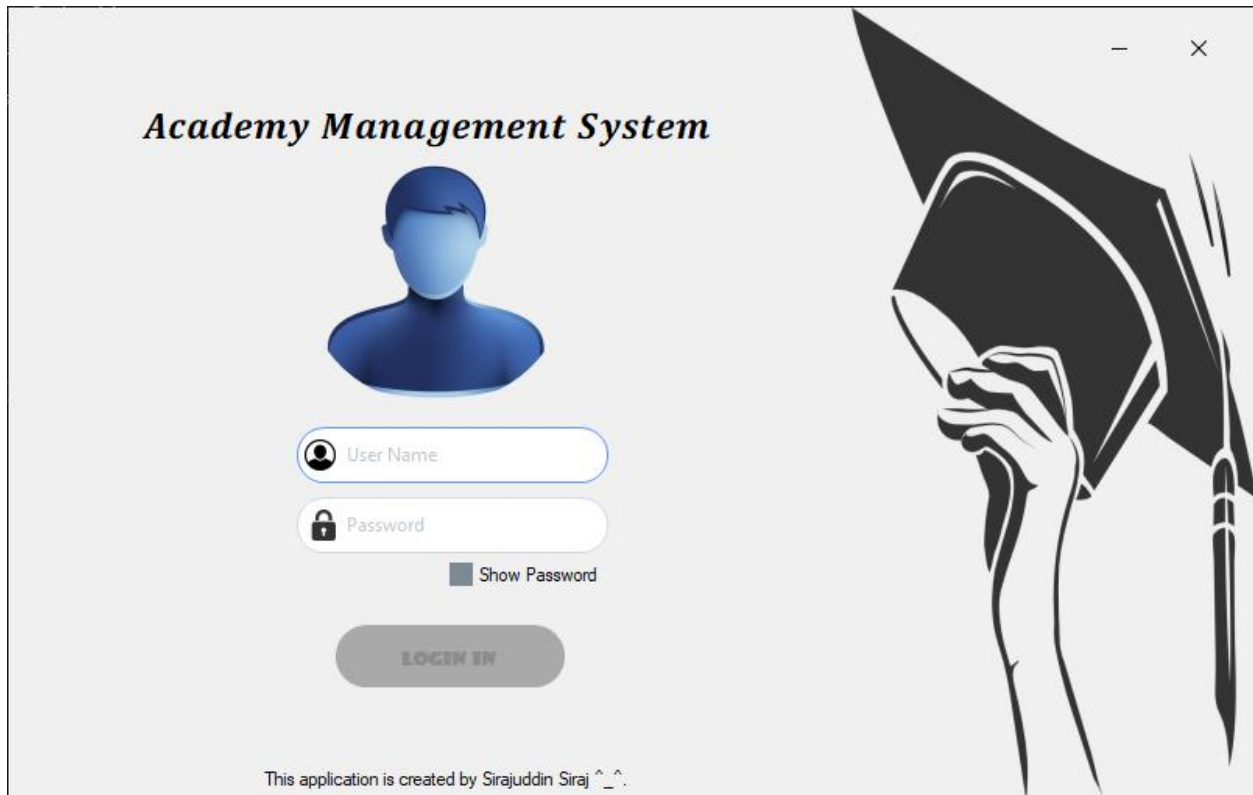
- a. Only a small number of possible inputs can be tested and many program paths will be left untested.
- b. Without clear specifications, which are the situation in many projects, test cases will be difficult to design.
- c. Tests can be redundant if the software designer/developer has already run a test case.
- d. Ever wondered why a soothsayer closes the eyes when foretelling events? So is almost the case in Black Box Testing.

*Chapter 4*

## **4 User Interface**

## 4.1 Login Form

Login form is used to authenticate user to the system.



The image shows a login form for the 'Academy Management System'. The form is displayed in a window with a light gray background. At the top, the title 'Academy Management System' is written in a bold, italicized font. Below the title is a blue silhouette of a person's head and shoulders. Underneath the silhouette are two input fields: 'User Name' with a person icon and 'Password' with a lock icon. To the right of the password field is a 'Show Password' checkbox. Below these fields is a 'LOGIN IN' button. On the right side of the window, there is a large, stylized illustration of a hand holding a graduation cap. At the bottom of the window, a small text line reads: 'This application is created by Sirajuddin Siraj ^\_^'.

Figure 4-1 Log in form

## 4.2 Side bar

The system has the following menu sections that accessing each section.

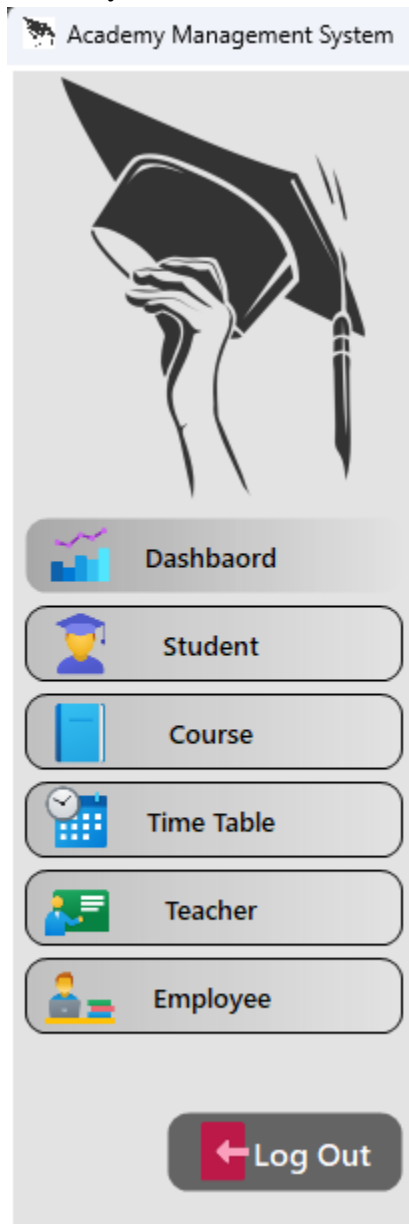


Figure 4-2 Side bar

## 4.3 Dashboard

In this page you can see the dashboard in which you can see, all the students, teacher and employees and also you can search the number of students in each semester by typing the number of semester..

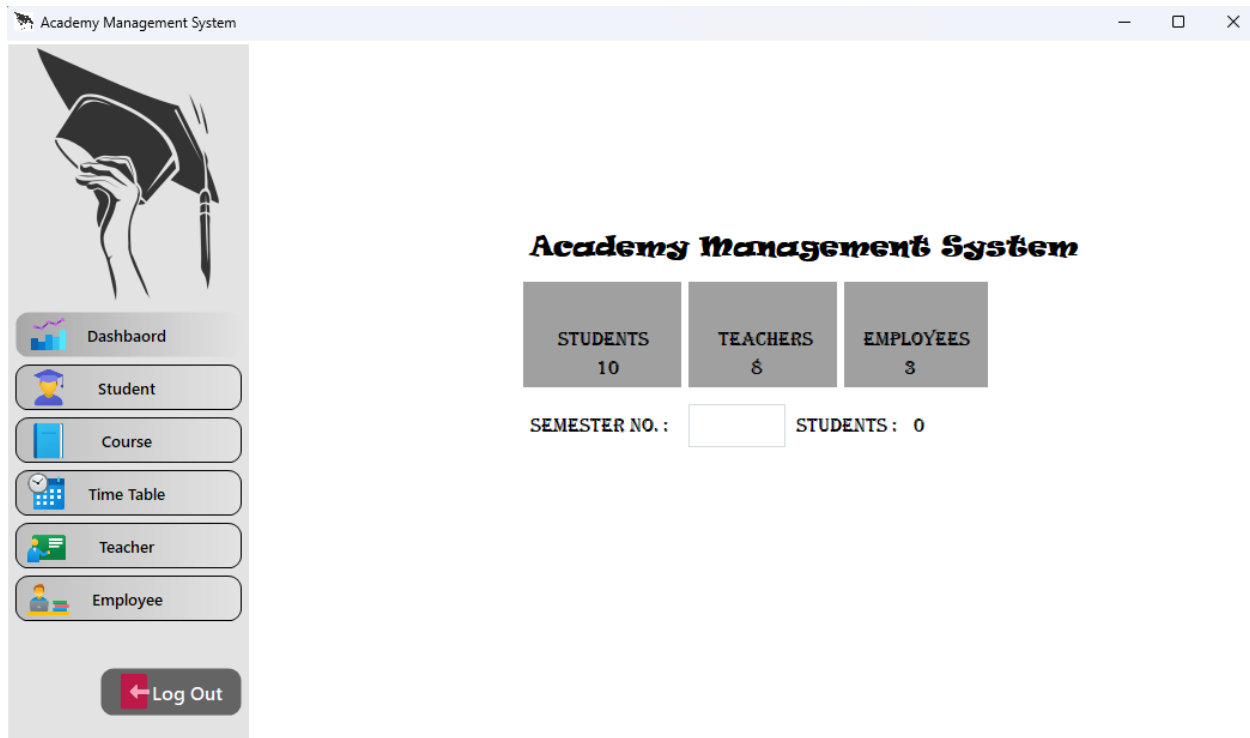




Figure 4-3 Dashboard page

## 4.4 Students

In this page you can Insert, delete, update, search and get a report of all the students.

 Academy Management System
 — □ ×



Dashbaord

Student

Course

Time Table

Teacher

Employee

Log Out

Student Entry

ID :

Gender :

Male

Name:

Course:

Level One

F/Name:

Time :

Phone No. :

Payment :

Address:

Remains :





0

Semester :

Duration:

0 Day(s)

Action Buttons

Student Search

Search ID :

Search Type:

ID

Students

std ID	Student Name	fname	phonenumber	address	gender	payment	remain	time
1	Fraidoon	Muhammad	Yosuf 783434	Enjil	Male	850	0	1:15 - 2:45 pm
2	yaqoob	Muhammad	Yosuf 783434	Enjil	Male	950	0	8:30 - 10:00 a
3	Neman	Jamil	783434	Enjil	Male	950	0	8:30 - 10:00 a
4	Omar	Taha	783434	Enjil	Male	1100	0	7:00 - 8:30 am
5	Omar	Taha	783434	Enjil	Male	1100	0	10:00 - 11:30 a

Figure 4-4 Students

## 4.5 Course

In this page you can Insert, delete, update, search and get a report of all the courses.

Academy Management System

**Add Course**

Search :

Course ID :

Title :

Duration :

Payment :

**Action Buttons**

+ (Add), ↺ (Update), ✖ (Delete), 📄 (View Details)

**Courses**

id	course_id	course_name	duration	payment	Select
8	1	Level One	60 days	950	Select
9	2	Level Two	60 days	1050	Select
11	3	Level T...	60 days	1100	Select
12	4	Level Four	60 days	1150	Select
13	5	Intensive	60 days	1200	Select
14	6	Advance 1	60 days	1600	Select
16	7	Advance 2	75 days	2200	Select
17	8	Advance 3	90 days	3300	Select
18	9	Beginner	60 days	850	Select

**Stack of Books:** 7th Styles, PHILOSOPHY, NICKY JAMES, PSYCHOLOGY, The Lost Mind, ADSTHETICS, YOUR 24 HOURS, THE SUPERMAN, GOOD vs BAD

Figure 4-5 courses Page

## 4.6 Time Table

In this page you can Insert, delete, update, search and get a report of all the time tables.

Academy Management System

**Time Table**

**Add Time**

Search Time:

Time ID :

Time :

**Action Buttons**

+

↺

✖

Aa

☰

Time Table			
id	time_id	time	Select
16	1	7:00 - 8:30 am	Select
18	2	8:30 - 10:00 am	Select
19	3	10:00 - 11:30 am	Select
20	4	11:30 - 1:00 pm	Select
21	5	1:15 - 2:45 pm	Select
23	6	2:45 - 4:15 pm	Select
24	7	4:15 - 5:45 pm	Select

Dashboard

Student

Course

Time Table

Teacher

Employee

Log Out

Figure 4-6 time table



## 4.7 Teacher

In this page, you can insert, delete, update or print a report of the teachers.

Academy Management System

Dashboard Student Course Time Table Teacher Employee Log Out

Add New Teacher Give Courses to Teacher

Add Teacher

ID :

Name:

F/Name:

Phone No. :

CNIC No. :

Address:

Status :

Action Buttons

Search

Search  Search Type

Teachers

id	Teacher_ID	Name	FatherName	Address	PhoneNumber	Cnic	Subject	Select
2	1	Ahmad Raziq	Wasiullah	Mazar Sh...	783434342	24rafa2e	Computer	Select
5	2	Ghulam F...	Ghulam Y...	Herat	783434342	24rafa2e	English	Select
6	3	Parwiz A...	hamidullah	Farah	783434342	24rafa2e	Arabic	Select
7	4	Shabir A...	Milad	Farah	783434342	24rafa2e	English	Select
8	5	Jalil Ahmad	Bashir A...	Farah	783434342	24rafa2e	Arabic	Select
9	6	Haronn	Khalil A	Nandahar	783434342	24rafa2e	Arabic	Select

Figure 4-7 Teacher

## 4.8 TeacherCourse

When a new Student joins the academy and the time and course, he/she selects here the teacher can be assign to that time and course. And a CRUD operation will be involved.

The screenshot displays the 'Academy Management System' interface. On the left is a sidebar with navigation buttons: Dashboard, Student, Course, Time Table, Teacher, Employee, and a Log Out button. The main content area has two tabs: 'Add New Teacher' and 'Give Courses to Teacher' (which is active). Under the 'Give Courses to Teacher' tab, there is a 'Give Course' form with three dropdown menus: 'Teacher' (selected: Ahmad Raziq), 'Course' (selected: Level One), and 'Time' (selected: 7:00 - 8:30 am). Below the form is a search section with a text input and a 'Search Type' dropdown (selected: ID). To the right of the form is an 'Action Buttons' panel containing icons for adding, refreshing, deleting, and editing records. At the bottom, a table titled 'List all Given Courses' shows the following data:

id	TeacherName	CourseName	Time
1052	Ahmad Raziq	Level One	8:30 - 10:00 am
1056	Ahmad Raziq	Level Three	7:00 - 8:30 am
1057	Ahmad Raziq	Level Three	10:00 - 11:30 am
1058	Jalil Ahmad	Level Three	10:00 - 11:30 am
1059	Parwiz Ahmad	Level Three	10:00 - 11:30 am
1060	Shabir Ahmad	Advance 1	7:00 - 8:30 am

Figure 4-8 TeacherCourse

### 4.8.1 Employee

This page shows a list of employees and you can try CRUD operation on them too.

Academy Management System

**Add Employee**

ID :

Name:

F/Name:





Phone No. :

CNIC No. :


Address:

Status :

**Action Buttons**

**Employees**



**Search**

Search :  Search Type :

Id	empID	empName	empFName	empPhone	empCnic	empStatus	empAddress	Select
11	2	Kamran	Perwiz	7834334	2382	Office W...	Herat	<input type="button" value="Select"/>
12	3	Bilal	Faridin	7834334	2382	Cleaner	Paktika	<input type="button" value="Select"/>
13	4	Lal Muha...	Jalil Ahmad	7834334	2382	Cleaner	Mazar Sh...	<input type="button" value="Select"/>

Figure 4-9 Employee List

*Chapter 5*

## 5 Conclusion and Future Work

### 5.1 Conclusion

While developing the system a conscious effort has been made to create, using many available tools, techniques, and resources to develop a proper desktop application that solves all the problems that academies faced with. Cost-effectiveness, user-friendliness, flexibility, multi-user, and multi-currency are the system uniqueness that distinguishes between other applications.

### 5.2 Future Work

The scope of this project includes what all future enhancements can be done in this system to make it more feasible. Here, I will write down the most important enhancement features that are required in the upcoming version of this system to be implemented.

- Make it more user-friendly and easier to access
- Add multi-language feature
- Dynamic multi-currency that user can add new currency in GUI
- Dynamic branches that user can add new branch in the system
- Dynamic access privileges by end-user
- Database Backup and restore
- Ending a financial year

## REFERCES

1. <https://www.c-sharpcorner.com>
2. <https://www.stackoverflow.com>
3. <https://www.w3school.com>
4. <https://www.tutorialspoint.com>
5. SQL server inside out Kindle([www.ebook-dl.com](http://www.ebook-dl.com))books