SOMAVILLE UNIVERSITY

DESIGN AND IMPLEMENTATION OF STUDENT RECORD MANAGEMENT SYSTEM

MOGADISHU SOMALIA

By

1	Yahye iise Mohamed	7783
2	Abdi salaam Ahmed Mohamed	10489
3	Abdullahi Awil Mohamed	8511
4	Khaalid nabiil Ali	8417
5	Yusuf Guled Mohamed	8831
6	Abdullahi Ahmed Hayow	8769
7	Abdullahi Farah Ali	9112
8	Abdulkadir Mohamed Abdule	8591
9	Abdirahman Ahmed Alasow	8965

A THESIS REPORT SUBMITTED TO *STUDENT RECORD MANAGEMENT SYSTEM*

in PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF

BACHELOR OF COMPUTER SCEINCE OF SOMAVILLE UNIVERSITY

July, 2023

DECLARATION

This thesis is our original work and has not been presented for a degree in any other university."

Yahye iise Mohamed

Signature ……..

Date ……..

Abdi salaam Ahmed Mohamed

Signature ……..

Date ……..

Abdullahi Awil Mohamed

Signature ……..

Date ……..

Khaalid nabiil Ali

Signature ……..

Date ……..

Yusuf Guled Mohamed

Signature ……..

Date ……..

Abdullahi Ahmed Hayow

Signature ……..

Date ……..

Abdullahi Farah Ali

Signature ……..

Date ……..

Abdulkadir Mohamed Abdulle

Signature ……..

Date ……..

Abdirahman Ahmed Alasow

Signature ……..

Date ……..

APPROVAL

We certify that this Research report satisfies the partial fulfillment of the requirements for the award of the Bachelor Degree of Computer Science & it at Somaville University in Mogadishu Somalia, and has been carried out by the candidates under our supervision and was submitted with our approval

SUPERVISOR: ABDIRISAK IBRAHIM ABDIWAHAB

Signature ……..

DATE ……...

DEDICATION

We dedicate this piece of work to our parents they have been source of inspiration, engine of courage and secret of our achievements. We also dedicate it to our sisters and brothers for all the support.

ACKNOWLWDGEMENT

We wish to acknowledge with profound gratitude to our supervisor Eng. **ABDIRISAK IBRAHIM ABDIWAHAB**, for the tireless support he has provided to during the preparation and execution of this work.

We are also genuinely grateful to the entire staff of Somaville University for their generous support and guidance during our studies. Special thanks go to the lectures in the "*STUDENT RECORD MANAGEMENT SYSTEM"* for introducing us to this very important and interesting discipline of computer science.

We would like to send most sincerely thank our classmates for the moral support and encouragement that they accorded to during the program.

Finally, we indebted to us fierily for their love, patience, and support during the entire time of us studies.

Allah blesses you all.

Table of Contents

CHAPTER ONE

1.0. INTRODUCTION

The registrar of Student Record Management System is responsible for handling student information and gathering them during enrollment.

This information includes each student's background information, student courses taken student attendance at lectures grades, performance record, and other information needed by the Institution.

1.1 BACKGROUND OF THE PROJECT

The background of student record management systems can be traced back to the emergence of computer technology and the need for educational institutions to streamline their administrative processes. In the past, student records were primarily maintained on paper, which made it difficult to organize and retrieve information quickly. With the advent of computers and databases, schools began to explore digital solutions to manage student data more effectively.

Early versions of student record management systems were simple and focused on basic functionalities like storing student names, addresses, and grades. However, as technology advanced, these systems evolved to include more comprehensive features and capabilities. Today, modern student record management systems are sophisticated software applications that integrate various modules and functionalities to support multiple aspects of student administration

1.2. PROPLEM STATEMENT

The current education system generates a massive amount of student data that needs to be efficiently managed and maintained. However, traditional methods of record-keeping, such as paper-based systems or disconnected spreadsheets, pose numerous challenges and limitations. These include:

1. Data Disorganization: Managing student records manually leads to disorganized data, making it difficult to find specific information in a timely manner. This can result in delays and errors in administrative processes, such as enrolment, grading, and generating reports.

2. Limited Accessibility: Paper-based records or local spreadsheet systems restrict access to authorized personnel only, making it inconvenient for multiple stakeholders, such as teachers, administrators, parents, and students, to access and update student information. This lack of accessibility hampers collaboration and efficient communication.

3. Data Security Risks: Storing student records in physical files or local systems makes them vulnerable to loss, damage, or unauthorized access. The mishandling of sensitive student information, such as grades, attendance, and personal details, can lead to privacy breaches and legal implications.

1.3. RESEARCH QUESTIONS

1. How to identify the usage of Student Record Management System?

2. How to analysis the requirements of Student Record Management System?

3. How to design and implementation of Student Record Management System?

1.4. PURPOSE OF THE PROJECT

1. to identify the usage of Student Record Management System?

2. to analysis the requirements of Student Record Management System?

3. to design and implementation of Student Record Management System?

1.5. PROJECT OBJECTIVIES

The main objective is to develop a robust Student Record Management System for
Specific Objectives;

i. To implement the system
ii. To test and validate the system
iii. To analysis the requirement

1.6 SCOPE OF THE PROJECT

a. Time scope should be march 15 up to jully 2023
b. Content scope of project Student Record Management System , Front end :HTML, CSS and javascript and Back end : PHP and MySql

1.7. SIGNIFICANCE OF THE PROJECT

A Student Information System is a major software solution that helps with everything from admissions to student success. It facilitates and manages key administrative processes,

maintains order and organization within a school's records, and can make universities and colleges more efficient.

1.8 ORGANIZATION OF THE PROJECT

his is systematic writing in this final thesis and will also consist of 6 chapters. Here is and explanation of each chapter, namely:

Chapter I-Introduction, in this section, we will deeply focus on the project background, statement of problems, objective of Project, purpose of project, scope of project questions, and significant of project

CHAPTER TWO

2.0 REVIEW OF RELATED LITERATURE

2.1 INTRODUCTION

A project on Student Information Management system which was carried out by Soita Reuben, an IT Student at Livingstone International University, It is developed to be used by Tertiary Institute to Maintain records of students easily. Achieving this objectives is difficult using a manual system as the Information is scattered, can be redundant and collecting relevant information can be very time Consuming. All these problems are solved using this project. Throughout the project the focus has Been on presenting information in an easy and intelligible manner. The project is very useful for those Who want to know about Student Information Management Systems and want to develop software on The same concept. The goal of this chapter is to explain the theory of database management system development which Will be applied in the development of student database management system for CCP vocational Institute. The following issues will be presented respectively. This will give an insight into the project area and help to get information that will enhance the Development of the student database management system.

2.2 CONCEPTS, OPINIONS, OR IDEAS FROM AUTHORS/ EXPERTS

Student record management systems have become increasingly popular in educational institutions in recent years. These systems are designed to simplify the process of storing, retrieving, updating, and managing the records of students. Several studies have been conducted on the use of student record management systems, and their impact on educational institutions.

One study by Hussain et al. (2018) explored the use of a student record management system in a Pakistani university. The study found that the system improved the efficiency of record management and reduced the workload of administrative staff. It also improved communication between students, faculty, and administrative staff, and provided students with easy access to their academic records.

Another study by Alansari et al. (2019) investigated the use of a student record management system in a Saudi Arabian university. The study found that the system improved the accuracy and completeness of student records, and reduced errors and inconsistencies in record

management. It also improved the efficiency of administrative processes and reduced the time and effort required to manage student records.

A study by Tladi et al. (2020) explored the use of a student record management system in a South African university. The study found that the system improved the transparency and accountability of record management, and provided students with easy access to their academic records. It also improved the efficiency of administrative processes and reduced the workload of administrative staff.

Overall, these studies suggest that student record management systems can have a positive impact on educational institutions. They can improve the efficiency and accuracy of record management, reduce errors and inconsistencies, improve communication between stakeholders, and provide students with easy access to their academic records. However, the success of these systems depends on their design, implementation, and user adoption

.2.3 PREVIOUS WORK/ EXISTING SYSTEMS

There are several existing student record management systems available in the market. Some of the most popular ones are:

1. PowerSchool: PowerSchool is a popular student information system used by K-12 schools and districts. It offers a suite of tools for managing student data, including attendance tracking, grade reporting, and parent communication.
2. Blackbaud Student Information System: Blackbaud is a popular student record management system used by private schools and independent schools. It provides tools for managing student data, enrollment, grade reporting, and more.

2.4 GAP ANALYSIS AND DIRECTION

A gap analysis is a technique used to determine the gap between the current state of a system or process and the desired state. In the case of a student record management system, a gap analysis would involve identifying the current state of the system and comparing it to the desired state. The desired state would be one that is efficient, effective, and meets the needs of all stakeholders involved in the system.

2.5 CHAPTER SUMMARY

A student record management system is a software application that is used to manage and store student data, including personal information, academic records, and other important data. A

literature review of student record management systems reveals that these systems can provide a number of benefits to educational institutions.

including increased efficiency, improved data accuracy,and enhanced communication

CHAPTER THREE

3.0 REQUIREMENT ANALYSIS

3.1 INTRODUCTION

The goal of requirement analysis in the development of a student record management system is to identify the needs and expectations of stakeholders, including administrators, instructors, and students, and translate these needs into a set of functional requirements that the system must meet.

3.2 USER REQUIREMENT ANALYSIS

To conduct a user requirement analysis for a student record management system, it is important to identify the various types of users who will interact with the system. These may include:

1. Administrators: These users will be responsible for managing the overall system, including setting up user accounts, configuring system settings, and generating reports.
2. Instructors: These users will be responsible for managing courses and academic progress for individual students, including grading assignments, tracking attendance, and communicating with students.
3. Students: These users will be the primary beneficiaries of the system, using it to access their academic records, view course materials, and communicate with instructors and administrators.

Based on the needs of these different user groups, the following user requirements may be identified:

1. User-Friendly Interface: The system should have a user-friendly interface that is easy to navigate, with clear menus and intuitive navigation. Users should be able to quickly and easily find the information they need, without having to click through multiple screens.
2. Customizable User Dashboards: Users should be able to customize their dashboards to display the information that is most relevant to them, such as upcoming assignments, course schedules, and academic progress.

3. Mobile Compatibility: The system should be compatible with mobile devices, allowing users to access their academic records and communicate with instructors and administrators from anywhere, at any time.
4. Communication Tools: The system should include tools for communication between students, instructors, and administrators, including the ability to send and receive messages and notifications.
5. Gradebook Management: Instructors should be able to manage their course gradebooks, including entering grades and comments, and generating reports on student performance.
6. Attendance Management: Instructors should be able to manage attendance for their courses, including recording and tracking student attendance over time.
7. Course Management: Instructors should be able to manage their courses, including creating new courses, assigning instructors, and tracking course progress.

8. Student Records: Students should be able to view their academic records, including grades, attendance, and other academic data.
9. Security and Access Control: The system should provide robust security and access control mechanisms to ensure that only authorized users can access sensitive data.
10. Reporting and Analytics: The system should provide comprehensive reporting and analytics capabilities, including the ability to generate reports on student performance, attendance, and other academic metrics.

3.3 PRELIMINARY INVESTIGATION

A preliminary investigation of a student record management system would likely involve assessing the current system in place (if there is one) and identifying any issues or areas for improvement. This may involve speaking with staff members who currently use the system, reviewing documentation and records, and observing the system in action.

From there, the investigation may involve researching potential software solutions, evaluating their features and capabilities, and determining whether they would be a good fit for the institution's needs. This may involve working with vendors or conducting demonstrations or trials of potential software solutions.

3.3.1 ORGANIZATION PROFILE

An organizational profile of a student record management system would typically include information about the educational institution that is using the system, as well as details about the system itself.

The educational institution's profile might include information such as its name, location, size, and student population. It might also include details about the types of programs and courses offered by the institution, as well as any specific needs or requirements that the student record management system must meet.

The profile of the student record management system might include information about the software vendor, the system's features and capabilities, and any customization options that are available. It might also include details about the system's implementation, including the timeline for deployment and any training or support that will be provided to staff members.

3.3.2 CURRENT SYSTEM

There are various student record management systems currently in use, but most of them have similar functionalities. Here are some common features of such systems:

1. Student information management: This includes basic information such as name, address, date of birth, contact information, and emergency contacts.
2. Enrollment management: This involves tracking the courses and programs that students are enrolled in, as well as their academic progress and performance.
3. Attendance tracking: This involves recording the attendance of students in various classes and activities.
4. Grade management: This includes recording and tracking the grades of students in various courses and programs.
5. Communication management: This allows teachers and administrators to communicate with students and parents through the system, sending messages, announcements, and updates.
6. Report generation: This involves generating various reports such as student transcripts, attendance reports, and grade reports.
7. Security and access control: This involves ensuring that the system is secure and that only authorized personnel have access to student information.

3.3.3 THE LIMITATIONS OF THE CURRENT SYSTEM

While student record management systems provide many benefits, there are also some limitations to consider. Some of these limitations include:

1. Lack of interoperability: Many student record management systems are not interoperable with other systems, which can make it difficult to share data between systems, causing data silos.
2. Security and privacy concerns: As student information is sensitive and confidential, there is always a risk of data breaches and unauthorized access. Inadequate security measures can put student data at risk.
3. Limited customization: Some systems may not allow for customization, which can make it difficult for institutions to tailor the system to meet their specific needs.
4. Cost: Implementing and maintaining a student record management system can be expensive, particularly for smaller institutions with limited budgets.
5. Learning curve: Introducing a new system can require a significant learning curve for staff and students, which can be challenging and time-consuming.
6. Inadequate support and training: Inadequate support and training can hinder the effective use of the system, making it difficult for staff and students to utilize its full potential.
7. Limited accessibility: Some systems may not be accessible to all users, particularly those with disabilities, which can create barriers to access and equity.

3.4 DATA GATHERING

Data gathering for a student record management system typically involves collecting and storing information about students, courses, grades, attendance, and other relevant data. Here are some of the common data gathering methods used in student record management systems:

1. Online registration: Students can register for courses online, providing personal information such as name, address, and contact details.
2. Student information system: This system is used to store and manage student data such as demographic information, academic history, and contact information.
3. Learning management system: This system is used to manage course materials, assignments, and assessments.

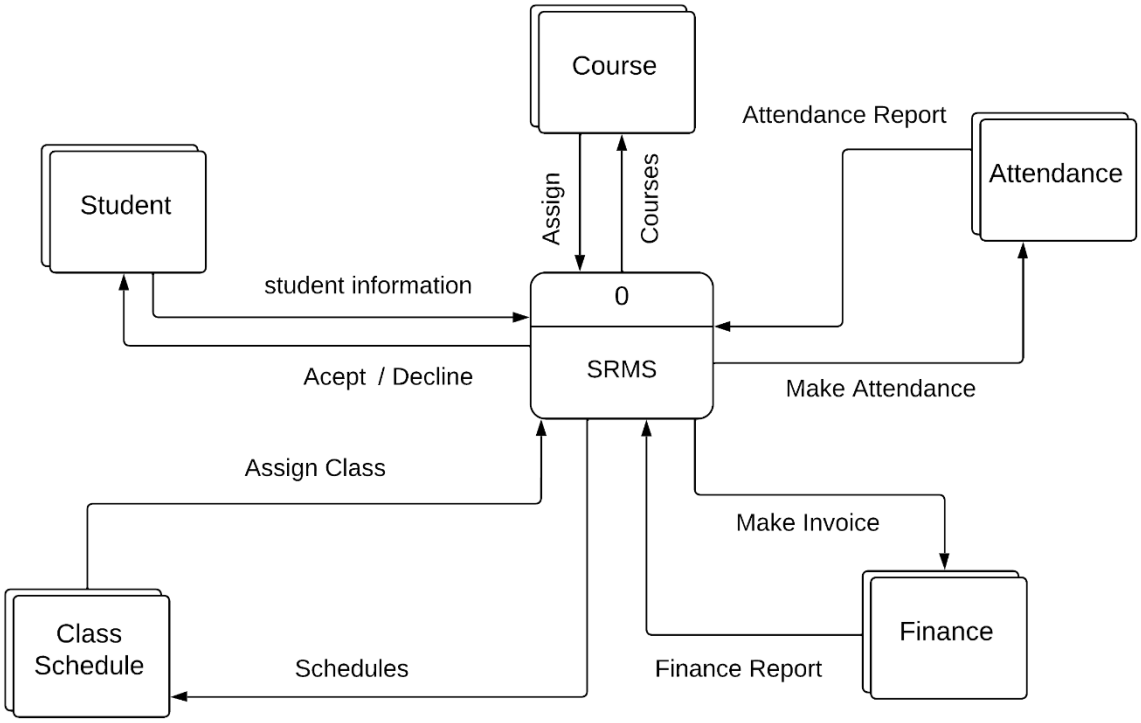
4. Attendance tracking system: This system is used to track student attendance in classes and other activities.
5. Gradebook system: This system is used to record and track student grades in various courses and programs.
6. Surveys and assessments: Surveys and assessments can be used to gather feedback from students about their experiences with the system, as well as their academic progress and performance.
7. Data integration: Various data sources can be integrated into the student record management system, such as financial aid data, student health data, and other relevant data.

3.4.1 DATA FLOW DIAGRAM (DFD)

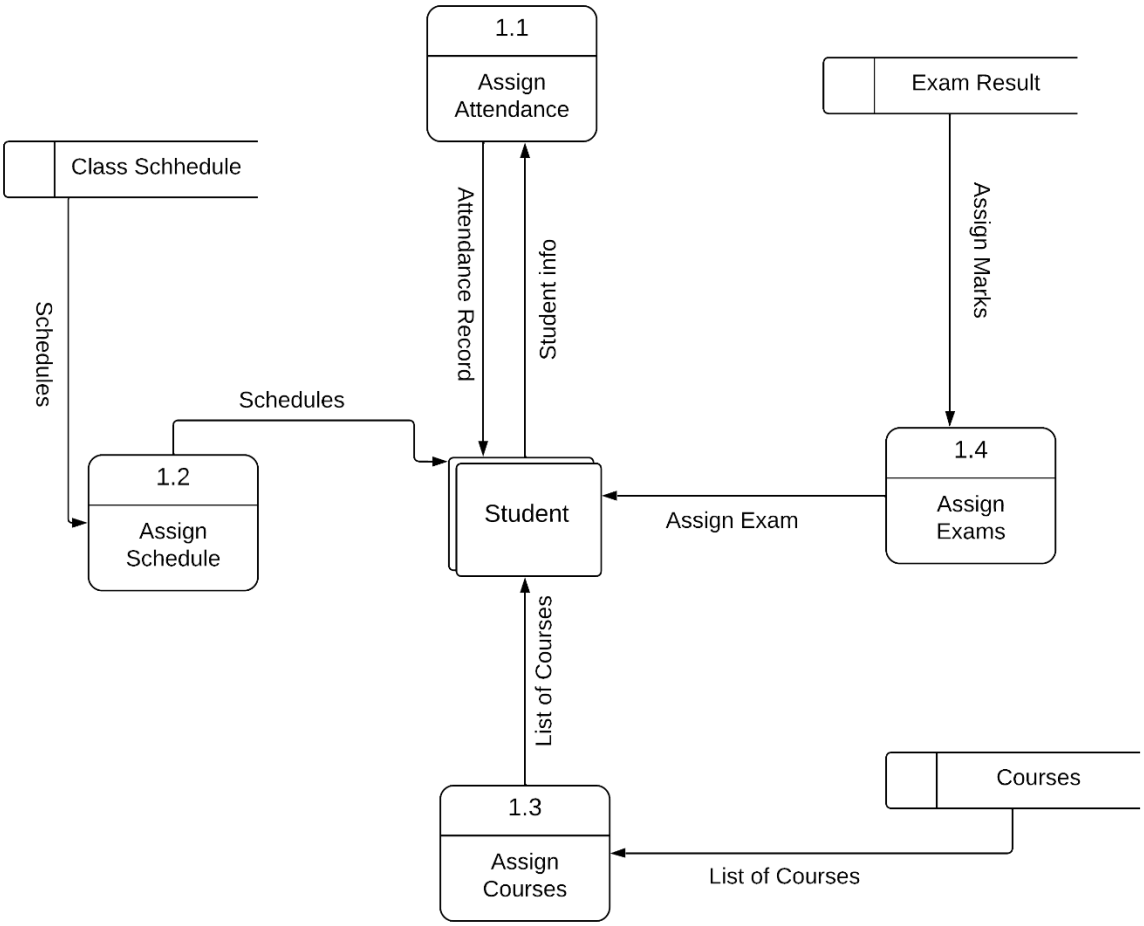
Student Information System Data flow diagram is often used as a preliminary step to create an overview of the Student without going into great detail, which can later be elaborated.it normally consists of overall application dataflow and processes of the Student process. It contains all of the userflow and their entities such all the flow of Timetable, Attendance, Class, Subject, Faculty, Semester, Student. All of **the below** diagrams has been used for the visualization of data processing and structured design of the Student process and working flow.

This is the Zero Level DFD of Student Information System,where we have eloborated the high level process of Student.It's a basic overview of the whole Student Information System or process being analyzed or modeled. It's designed to be an at-a-glance view of Faculty,Semester and Student showing the system as a single high-level process, with its relationship to external entities of Timetable,Attendance and Class. It should be easily understood by a wide audience, including Timetable,Class and Faculty In zero leve DFD of Student Information System, we have described the high level flow of the Student system.

Level Zero: Student Record Management System



Level One: Student Record Management System



3.5 PROBLEM STATEMENT

The problem statement for a Student Record Management System may include the following:

The traditional method of maintaining student records is time-consuming and prone to errors. It involves storing information manually in paper-based files, which can be difficult to update, retrieve, and share. This process can also result in data duplication and inconsistencies, making it challenging for academic departments to make informed decisions about student progress and performance.

To address these issues, there is a need for a Student Record Management System that can automate the process of storing, updating, and retrieving student information. The system should allow academic departments to manage student records efficiently, including personal information, academic progress, financial records, and course registration.

The Student Record Management System should also be accessible to authorized users, such as academic advisors, professors, and department heads, who need to access student records for various purposes. The system should provide a secure and reliable platform for managing and sharing student information, while also ensuring data privacy and protection.

3.6 FEASIBILITY STUDY

With the continuous growth and development of society, computer technologies and Informational systems are becoming increasingly prominent resources in many aspects of life relating the decision to implement any new project or program must be based on a thorough analysis of current operation system. The objective of feasibility study to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below

a. To analyze whether the software will meet organizational requirements.

b. To determine whether the software can be implemented using the current technology and within the specified budget and schedule.

c. To determine whether the software can be integrated with other existing software

There are the various types of feasibility that are commonly considered as the following steps

1. Technical feasibility study

2. Operational feasibility study

3. Economic feasibility study

4. Report feasibility study

5. Schedule feasibility study.

3.6.1. TECHNICAL FEASIBILITY STUDY

Technical feasibility Study assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget.

Table 3.6.1.1 TECHNICAL FEASIBILITY STUDY OF SOFTWARE AND HARDWARE

No	Item	Description	Amount
1	Proccessor	2.4 GHZ processor speed	200.00
2	Memory	128 GB RAM(16 GB Recomended)	100.00
3	Disk space	120 GB (including 20 GB for database Management System)	250
4	Display	800 x 600 colors (1024 x 768 high color 16 bit Recommended	40
5	Operating system	Window 7 or later	40
6	Server	Apacha server	300
Total			930

3.6.2. OPERATIONAL FEASIBILITY STUDY

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed.

3.6.2.1 OPERATIONAL HUMAN RESOURCES

No. of developers	Developer's job	Time	Cost	Total
4	Analysis phase	Six Weeks	70	70
3	Design phase	four Weeks	50	50
5	Configuration phase	three Weeks	90	90
2	The Thesis	two Weeks	80	80
Total				$290

3.6.3. ECONOMIC FEASIBILITY STUDY

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development.

Table 3.6.3.1 ECONOMIC DEVELOPING COST EXPENSED

No	Terms	Cost	Total
1	Technical cost	930	930
2	Operational cost	290	290
3	Development cost	200	200
Grand Total			1,420

3.6.4. REPORT FEASIBILITY STUDY

Feasibility Study is an assessment of the practicality of a proposed project or system related the sequence of work flow of frame-working by considering the reports of Total cost ownership (TCO) including the reports gained from technical feasibility study, operational feasibility, economic feasibility, and schedule feasibility where its estimated financial cost budget allocating for operating expenses and development costs, new recruitment staff , IT experts and consultants are part of cash flow which expensed and including software, hardware, training programs mentioned above table

3.6.5 SCHEDULE FEASIBILITY STUDY

Schedule feasibility is the degree to which a deadline for a strategy, plan, project or process is realistic and achievable

Operationally started in 02/12/2022 and will be completed if Allah says 05/07/2023

Task No	Activity	Duration Per Week	Percentage
A	Analysis	1 week	25%
B	Design	3 week	15%
C	Code	1 week	50%
D	implementation	1 week	10%

3.7 USER REQUIREMENT SPECIFICATION

Here is a possible USER REQUIREMENT SPECIFICATION for a student record management system:

1. Introduction:

The Student Record Management System is a web-based application designed to manage the records of students in an educational institution. The purpose of the system is to simplify the process of storing, retrieving, updating, and managing the records of students.

2. User Roles:

The system will have the following user roles:

- Administrator: responsible for managing the system and its users.

- Faculty: responsible for managing the records of students in their respective courses.

- Student: responsible for accessing their own records.

3. Functional Requirements:

The system should provide the following functionalities:

- User Authentication: The system should allow users to login with their credentials.

- Student Registration: The system should allow the administrator to register new students with their personal and academic information.

- Course Management: The system should allow the administrator to manage the courses offered by the institution.

- Student Record Management: The system should allow the faculty to manage the records of students enrolled in their respective courses, including grades, attendance, and other academic information.

- Report Generation: The system should allow the administrator and faculty to generate reports on student performance, attendance, and other academic information.

- Search and Filter: The system should allow users to search and filter student records based on various criteria such as name, course, and academic performance.

4. Non-Functional Requirements:

The system should meet the following non-functional requirements:

- Scalability: The system should be scalable to accommodate a large number of students and users.

- Security: The system should ensure the security and confidentiality of student records and user information.

- Accessibility: The system should be accessible from any device with an internet connection.

- Performance: The system should have a fast response time and be able to handle multiple users simultaneously.

5. Constraints:

The system should be developed using modern web development technologies and should be compatible with popular web browsers. The system should also comply with relevant data protection and privacy laws.

6. Assumptions:

The system assumes that students, faculty, and administrators have basic computer literacy and internet connectivity. The system also assumes that the institution has an existing database of student information that can be imported into the system.

3.8 CHAPTER SUMMARY

In this chapter, we have analyzed the user requirement in two kinds the first is user requirement Analysis that discussed the problems and limitations for the current system, why the user needs to get student record management System that had presented in its preliminary investigation for Organizational profile and after all that we have proposed the solution of that limitations that is to ensure the design and implementation student record management system activities that clearly and securely shown mentioned above the sections technique method of analysis tool modeling the user requirement at end of this chapter, will define the specifically secure Architecture diagram implement for design phase about proposed system

CHAPTER FOUR

SYSTEM DESIGN

4.1 INTRODUCTION

In chapter three of this book we have conducted a detailed requirement analysis for the old system and requirements of the new proposed system which is a prerequisite for the design phase. So, in this chapter our main focus is the systems design which is one of the most important phases in

the systems development life cycle (SDLC). Systems design is a broad term that encompasses data design, user interface design and report design. All these design types will be covered in this chapter in detail. Design requires extreme carefulness; because errors in the design phase can cause a failure of the whole system or project

4.2 DESIGN GOALS

There are many aspects to consider in the design of an every system project. The importance of each should reflect the goals the system is trying to achieve. Some of these aspects are:

- **Efficiency** – It is generally considered to be the most important. Given a piece of hardware on which the system will run and a piece of software to run it, the design should make full and efficient use of the facilities provided. The users should interact with the system without any delay.

- **Integrity**- this means accuracy in simple terms. During design, integrity always related to data. Integrity is ensured when data are not corrupted by hardware or software and data should be maintained; and accurately representing real world.

- **Compatibility** - The system software is able to operate with other products that are designed for interoperability with another product.

- **Modularity** –another term for modularity is a module or component. System based on modules makes understanding of the system easier, as consequence, reducing the cost to build the software.

- **Reliability -** The system is able to perform a required function under stated conditions for a specified period of time.

- **Reusability -** the system is capable to be enhanced and added further features and modification with slight or no modification at one of its major components.

- **Security –** there is always a security concern in every system. Therefore, the system should protect any harmfulness from the computer themselves and access from unauthorized users.

4.3 DATABASE DESIGN

Designing a Database Like most tasks, building a database starts with a design. After all, you wouldn't try to build a house without a blueprint, and most people wouldn't attempt to prepare a new dish without a recipe. Like these other tasks, having a good design for your database is a major first step in creating a successful project.

In designing a database application, you must set up not only the program's routines for maximum performance, but you must pay attention also to the physical and logical layout of the data storage. A good database design does the following:

- Provides minimum search times when locating specific records.
- Stores data in the most efficient manner possible to keep the database from growing too large.
- Makes data updates as easy as possible.

4.4 ENTITY RELATIONSHIP DIAGRAM (ERD)

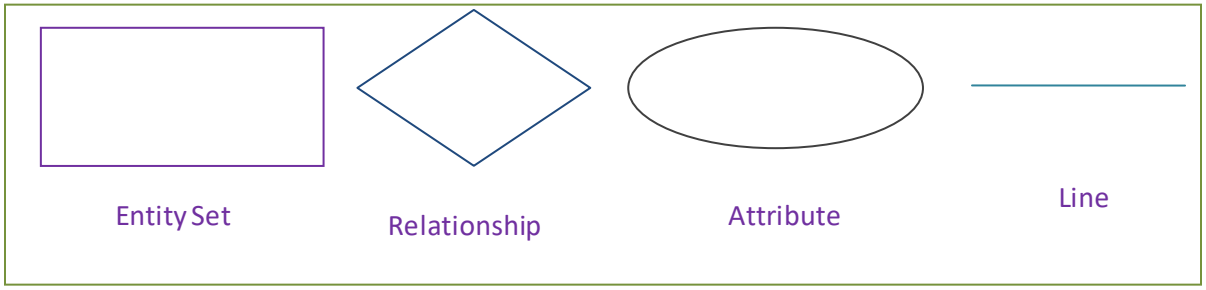
ERD is a detailed, logical representation of the entities, associations and data elements for an organization or business. **ERD** is a graphical modeling tool to standardize ER modeling; the modeling can be carried out with the help of pictorial representation of entities, attributes and relationships.

The basic building blocks of ERD are Entity, Attributes, Relationship and lines entity is an object Entity types or Entity set is a collection of similar entities; an entity may belong to more than one entity type. A relationship is an association of entities where the association includes one entity From other particular types is meaningful association before entity types. Attributes are properties of entity types in other words; entities are described in a data base by a set of attributes.

ERD Symbols

ER diagram is used to represent database schema.

- A rectangle represents an entity set. A Diamond represents Relationship An ellipse represents an attributes.
- Lines represent linking of attributes to entity sets & of entity sets to relationship sets.



Entity Relationship Diagram Basic Symbols

Relationship classification

Relationship is an association among one or more entities. This relationship can be broadly classified into one-to-one relation, one-to-many relation, many-tomany relation and recursive relation.

- **One to many Relationship Type:** The relationship that associate one entity to more than one entity is called one to many relationship: - Example is country having states for one country there can be more than one states hence is an example one to many relationship.
- **One to one Relationship Type**: One to one relationship is a special case of one to many relationships. True one to one relationship is rare. The relationship between the president and Country is an example is one to one relationship.
- **Many to Many Relationship Type**: The relationship between EMPLOYEE entity and PROJECT entity is an example of many to many

relationships. Many employees will be working in many projects hence the relationship between employee and project is many to many relationships.

- **Many to One Relationship Type**: The relationship between EMPLOYEE and DEPARTMENT is an example of many to one relationship, there may be EMPLOYEE working in one DEPARTMENT. Hence relationship between EMPLOYEE and DEPARTMENT is many to one relationship.

Table2.1 Relationship types

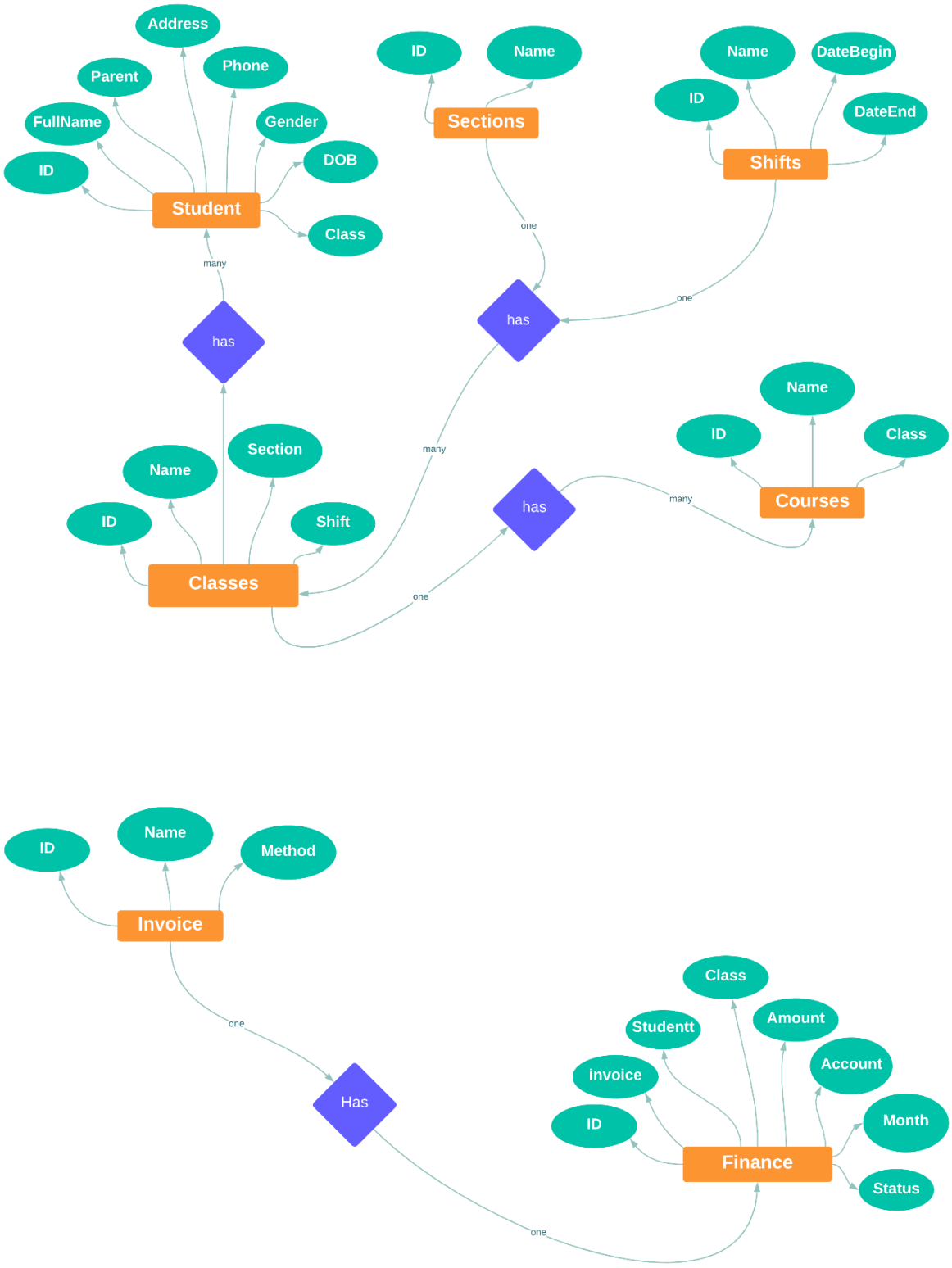
Relation Type	Representation
One-to-one	
One-to-many	
Many-to-many	
Many-to-one	

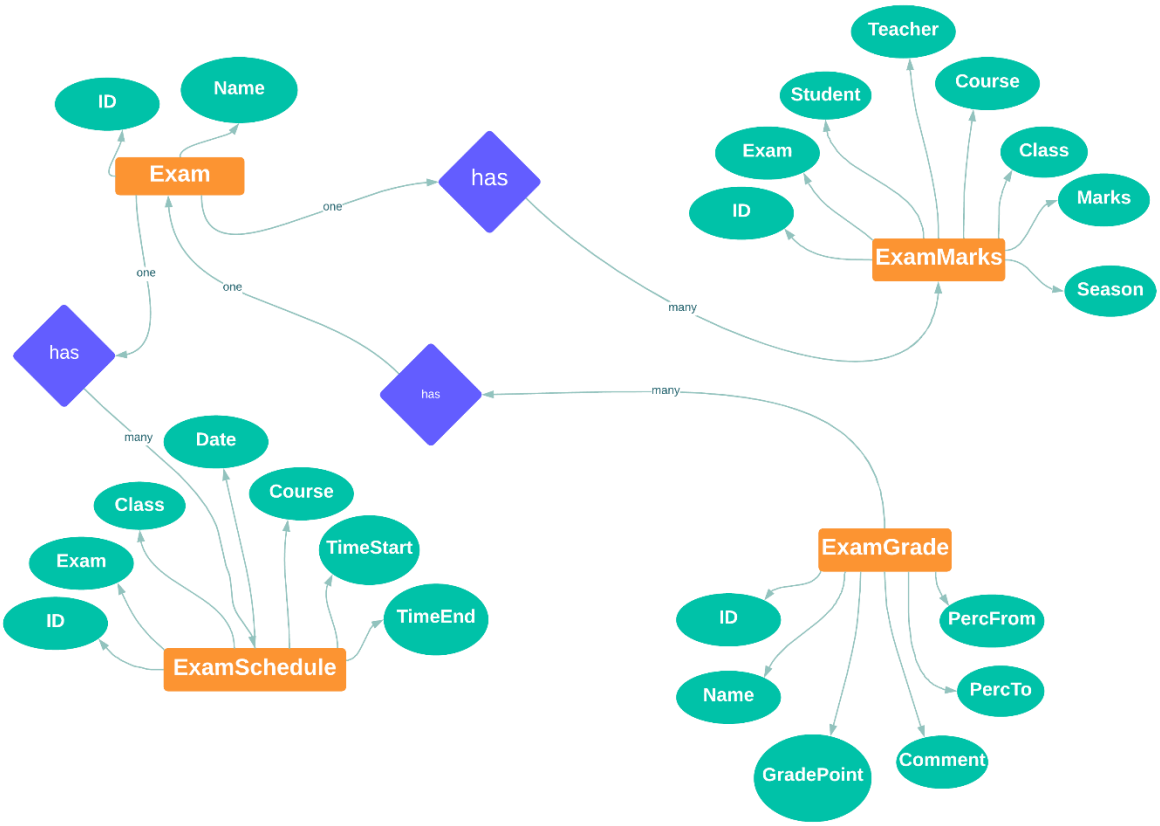
4.4.1 De-normalization

In a relational database, demoralization is an approach to speeding up read performance (data retrieval) in which the administrator selectively adds back specific instances of redundant data after the data structure has been normalized. A demoralized database should not be confused with a database that has never been normalized.

During normalization, the database designer stores different but related types of data in separate logical tables called relations. When a query combines data from multiple tables into a single result table, it is called a join. Multiple joins in the same query can have a negative impact on performance. Introducing demoralization and adding back a small number of redundancies can be a useful for cutting down on the number of joins.

Entity relationship




ID
Name
Exam
has
one
Teacher
Course
Student
Exam
Class
ID
Marks
ExamMarks
Season
one
one
has
many
has
many
many
Date
Class
Course
Exam
TimeStart
ID
TimeEnd
ExamSchedule
ExamGrade
ID
PercFrom
Name
PercTo
GradePoint
Comment

4.4.2 NORMALIZATION

Normalization is the process of decomposing large tables into smaller ones in order to eliminate redundant data and duplicate data and to avoid problems with inserting, updating, or deleting data. During the normalization process, table structures are tested against normal forms and then modified if any of the aforementioned problems are found. A normal form is a specific set of rules that can be used to test a table structure to ensure that it is sound and free of problems. There are a number of normal forms, and each one is used to test for a particular set of problems. The normal forms currently in use are First Normal Form, Second Normal Form, Third Normal Form, Fourth Normal Form, Fifth Normal Form, Boyce-Codd Normal Form, and Domain/Key Normal Form. Normalization is the primary tool to validate and improve a logical design so that it satisfies certain constraints that avoid unnecessary duplication of data. A relational table is said to be a particular normal form if it satisfied a certain set of constraints.

Purpose of Normalization

Normalization allows us to minimize insert, update, and delete anomalies and help maintain data consistency in the database.

- To avoid redundancy by storing each fact within the database only once
- To put data into the form that is more able to accurately accommodate change
- To avoid certain updating "anomalies".
- To facilitate the enforcement of data constraint.
- To avoid unnecessary coding.

First Normal Form (1NF): A table is in first normal form (1NF) if and only if all columns contain only atomic values; that is, there are no repeating groups (columns) within row. It is to be noted that all entries in a field must be of same kind and each field must have a unique name, but order of field (columns) is irrelevant. Each record must be unique and the order of the rows is irrelevant.

Second Normal Form (2NF)

A table is in second normal form (2NF) if and only if it is in 1NF and every non key attributes is fully dependent on the primary key.

Third Normal Form (3NF)

To be in Third Normal Form (3NF) the relation must be in 2NF and no transitive dependencies may exist within relation. A transitive dependency is when as attributes is indirectly functionally dependent on key (that is, the dependency is through another non-key attributes).

4.5 Data Dictionary

Table 4.5.1 ***bank table***

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*
3	Acc_No	int(11)			No	*None*
4	Balance	decimal(10,2)			No	*None*

Table 4.5.2 ***attendance table***

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	AttendanceNo	int(11)			No	*None*
3	Student	int(11)			No	*None*
4	Class	int(11)			No	*None*
5	Course	int(11)			No	*None*
6	Teacher	int(11)			No	*None*
7	Status	tinyint(1)			No	0
8	Season	varchar(50)	utf8mb4_general_ci		No	*None*
9	Date	datetime			No	current_timestamp()
10	Taked	tinyint(1)			No	*None*

Table 4.5.3 ***classes table***

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*
3	Section	int(11)			No	*None*
4	Shift	int(11)			No	*None*

Table 4.5.4 ***class schedule table***

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Class	int(11)			No	*None*
3	Day	varchar(50)	utf8mb4_general_ci		No	*None*
4	Course	int(11)			No	*None*
5	TimeStart	time			No	*None*
6	TimeEnd	time			No	*None*
7	Season	int(11)			No	*None*

Table 4.5.5 courses ***table***

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*
3	Class	int(11)			No	*None*

Table 4.5.6 employees *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*
3	Address	varchar(50)	utf8mb4_general_ci		No	*None*
4	User	tinyint(1)			No	0
5	Phone	varchar(20)	utf8mb4_general_ci		No	*None*
6	Email	varchar(50)	utf8mb4_general_ci		No	*None*
7	Gender	varchar(10)	utf8mb4_general_ci		No	*None*
8	JobTitle	varchar(50)	utf8mb4_general_ci		No	*None*
9	Photo	text	utf8mb4_general_ci		No	*None*
10	Date	date			No	current_timestamp()
11	Status	tinyint(1)			No	0
12	Salary	decimal(10,2)			No	*None*

Table 4.5.7 exams *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*

Table 4.5.8 exam grade *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*
3	GradePoint	varchar(10)	utf8mb4_general_ci		No	*None*
4	percentageFrom	int(11)			No	*None*
5	percentageUpto	int(11)			No	*None*
6	Comment	varchar(50)	utf8mb4_general_ci		No	*None*

Table 4.5.9 exam marks *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	ExamNo	int(11)			No	*None*
3	Student	int(11)			No	*None*
4	Class	int(11)			No	*None*
5	Course	int(11)			No	*None*
6	Marks	int(11)			No	0
7	Exam	int(11)			No	*None*
8	Teacher	int(11)			No	*None*
9	Season	int(11)			No	*None*
10	Submited	tinyint(1)			No	*None*

Table 4.6.1 exam schedule *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Exam	int(11)			No	*None*
3	Class	int(11)			No	*None*
4	Course	int(11)			No	*None*
5	Date	date			No	*None*
6	TimeStart	time			No	*None*
7	TimeEnd	time			No	*None*

Table 4.6.2 finance *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	InvoiceNo	int(11)			No	*None*
3	Invoice	int(11)			No	*None*
4	Student	int(11)			No	*None*
5	Class	int(11)			No	*None*
6	User	int(11)			No	*None*
7	Amount	decimal(10,0)			No	*None*
8	Account	int(11)			No	*None*
9	Status	tinyint(1)			No	0
10	Date	date			No	current_timestamp()
11	Month	date			No	*None*
12	Season	int(11)			No	*None*
13	Phone	varchar(50)	utf8mb4_general_ci		No	*None*

Table 4.6.3 invoice *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*
3	Method	int(11)			No	*None*
4	Price	decimal(10,0)			No	*None*

Table 4.6.4 links *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	text	varchar(50)	utf8mb4_general_ci		No	*None*
3	icon	varchar(50)	utf8mb4_general_ci		No	*None*
4	link	varchar(50)	utf8mb4_general_ci		No	*None*

Table 4.6.5 parents *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	FullName	varchar(255)	utf8mb4_general_ci		No	*None*
3	Address	varchar(50)	utf8mb4_general_ci		No	*None*
4	Phone	varchar(50)	utf8mb4_general_ci		No	*None*
5	Email	varchar(50)	utf8mb4_general_ci		No	*None*
6	Occubation	varchar(50)	utf8mb4_general_ci		No	*None*
7	Photo	text	utf8mb4_general_ci		No	*None*
8	UserName	varchar(50)	utf8mb4_general_ci		No	*None*
9	Password	varchar(255)	utf8mb4_general_ci		No	*None*
10	Gender	varchar(10)	utf8mb4_general_ci		No	*None*
11	Status	tinyint(1)			No	1

Table 4.6.6 seasons *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*
3	Status	tinyint(1)			No	0

Table 4.6.7 sections *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*

Table 4.6.8 shifts *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*
3	DateBegin	time			No	*None*
4	DateEnd	time			No	*None*

Table 4.6.9 students *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	FullName	varchar(50)	utf8mb4_general_ci		No	*None*
3	Address	varchar(50)	utf8mb4_general_ci		No	*None*
4	Phone	varchar(50)	utf8mb4_general_ci		No	*None*
5	Class	int(11)			No	*None*
6	Parent	int(11)			No	*None*
7	Photo	text	utf8mb4_general_ci		No	*None*
8	Gender	varchar(10)	utf8mb4_general_ci		No	*None*
9	DOB	date			No	*None*
10	Status	tinyint(1)			No	1
11	UserName	varchar(50)	utf8mb4_general_ci		No	*None*
12	Password	varchar(255)	utf8mb4_general_ci		No	*None*

Table 4.7.1 student promotion *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Student	int(11)			No	*None*
3	ClassFrom	int(11)			No	*None*
4	ClassTo	int(11)			No	*None*
5	CurrentSeason	int(11)			No	*None*
6	PromotionSeason	int(11)			No	*None*

Table 4.7.2 teachers *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	Name	varchar(50)	utf8mb4_general_ci		No	*None*
3	Email	varchar(50)	utf8mb4_general_ci		No	*None*
4	Phone	varchar(50)	utf8mb4_general_ci		No	*None*
5	Address	varchar(50)	utf8mb4_general_ci		No	*None*
6	Gender	varchar(10)	utf8mb4_general_ci		No	*None*
7	Photo	text	utf8mb4_general_ci		No	*None*
8	Classes	varchar(255)	utf8mb4_general_ci		No	*None*
9	Courses	varchar(255)	utf8mb4_general_ci		No	*None*
10	Privileges	varchar(255)	utf8mb4_general_ci		No	*None*
11	UserName	varchar(50)	utf8mb4_general_ci		No	*None*
12	Password	varchar(255)	utf8mb4_general_ci		No	*None*
13	Status	tinyint(1)			No	1

Table 4.7.3 users *table*

#	Name	Type	Collation	Attributes	Null	Default
1	ID	int(11)			No	*None*
2	UserName	varchar(50)	utf8mb4_general_ci		No	*None*
3	Password	varchar(255)	utf8mb4_general_ci		No	*None*
4	Privileges	text	utf8mb4_general_ci		No	*None*

4.7 FORM DESIGN

Figure 4.7.1 Log in From

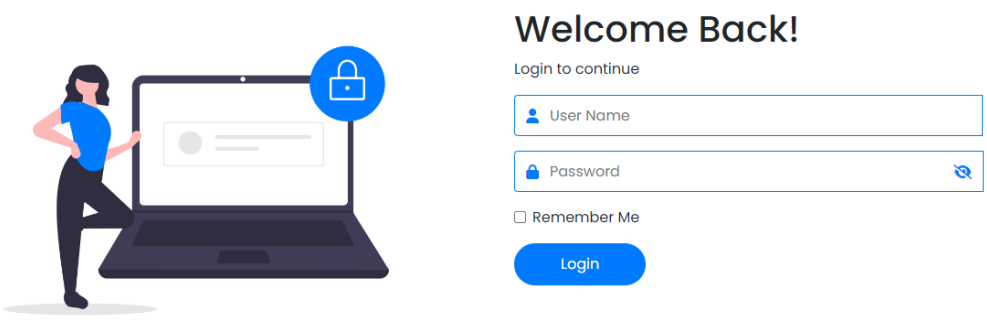


Figure 4.7.2 Home Page

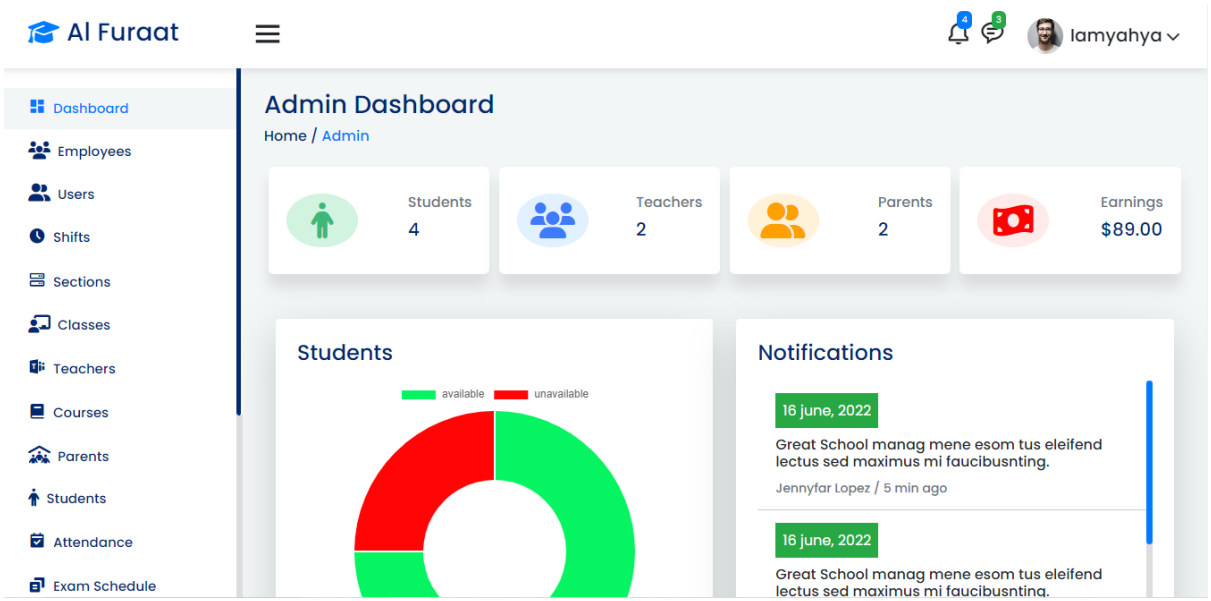


Figure 4.7.3 Registration From

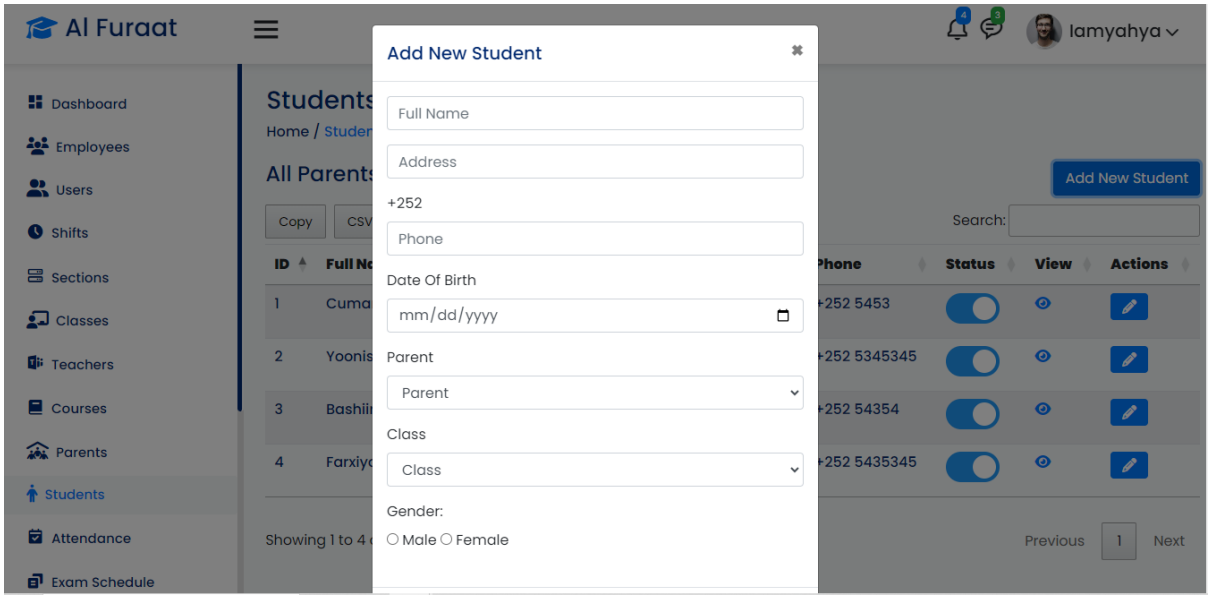
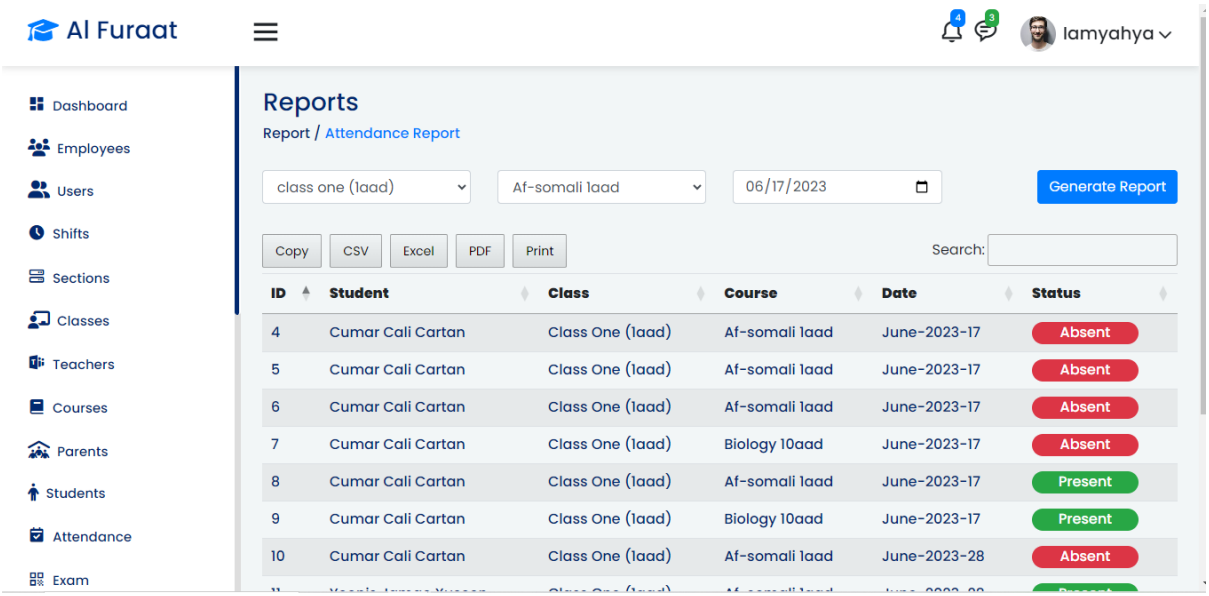


Figure 4.7.4 Report From



4.8 CHAPTER SUMMARY

System design being one of the most important chapters in developing systems. We talked about the design goals, normalization, and demoralization which are essential for designing robust database. The chapter also covered data dictionaries, table design, and form design

CHAPTER FIVE

SYSTEM DEVELOPMENT

5.0 INTRODUCTION

After coding, we tested each program to make sure that it is functioning correctly.
Later, programs are tested in groups, and finally the project is tested entirely.

The first step is to compile program using a case tool or language compiler

5.1 CODING PHASE

This phase is devoted to providing access to most of the computer programs that have been used to prepare the data and apply the programming techniques. Instructions and how to construct and install this application project and using the software included is made available on an attached compact disk (CD) that comes with this graduation book.

5.2 TYPES OF TESTING

A-Unit testing focus on verification effort on the smallest unit of software designs the software component or module. using the component level design description as a guide ,important control paths are tested to uncover errors within the boundary of module .the relative complexity of tests and the errors those tests the unit test

5.2.1 Unit testing

The testing of individual program or module is called unit testing. The objective is to identify and eliminate execution errors that could cause the program to terminate abnormally, and errors that could have been missed during the checking.

5.2.2 Integration testing

Integration testing is systematic technique for constructing the software architecture while at the same time conducting tests to uncover errors associated interface .the objective is to take unit tested components and build program structure that has been dictated by design. There is tendency to attempt no incremental integration; that is construct the program using a big bang

Incremental integration is the antithesis of big bang approach .the program is constructed and tested in small increments ,where errors are easier to isolate and correct ;interface are more

likely to be tested completely and a systematic test approach may be applied in the paragraphs that follow a number of different incremental integration strategies are discussed.

Top down integration testing is incremental approaches to construction of software module are integrated by moving downward through the control hierarchy. The Integration process is performed in a series of some steps

- The main control module is used as test driver and stubs are substituted for all components directly subordinate to the main control module
- Tests are conducted as each component is integrated
- On completion of each set of tests ,another stub is replaced with the real component

5.2.3 System testing

System testing is actually a series of different test whole primary purpose is to fully exercise the computer based system .although each test has a different purpose ,all work to verify that system element have been properly integration and pretest that are worthwhile for software based system

- Recovery testing
- Security testing
- Stress testing

Recovery testing is system test that forces the software to fail in variety of ways and verifies that recovery is properly performed .if recovery is automatic (performed by the system itself), Re-initialization, check pointing mechanisms.

Security testing attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration during security testing the tester plays the role(s) of individual who desires to penetration the system.

Stress testing executes a system in manner that demands resources in abnormal quantity, frequency or value

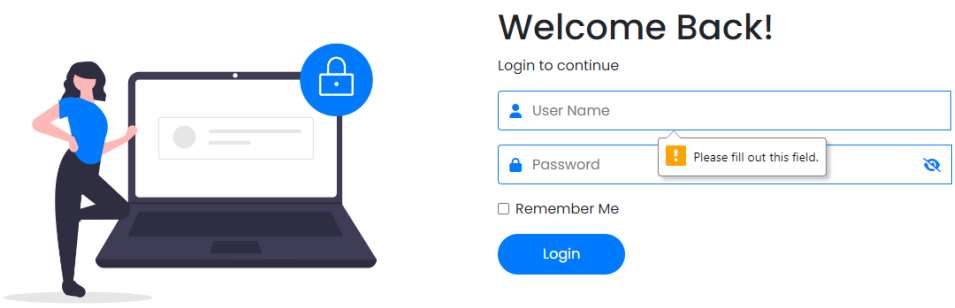


Figure 8.2.1 User info Fill Blank login

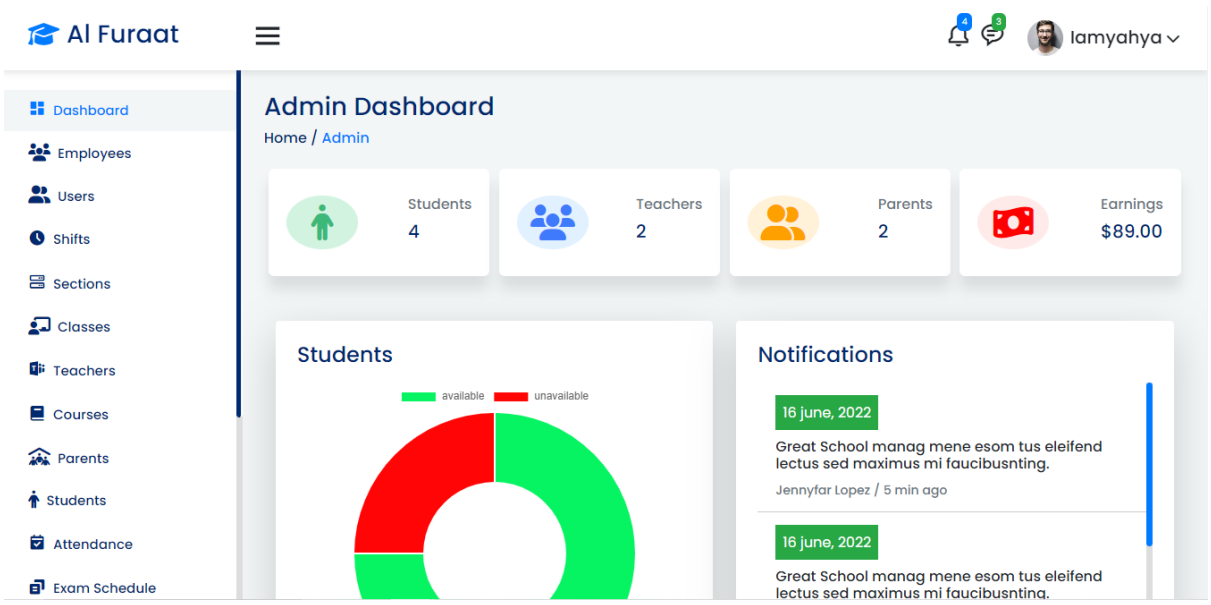


Figure 8.2.2 Home Page Student Record Management System

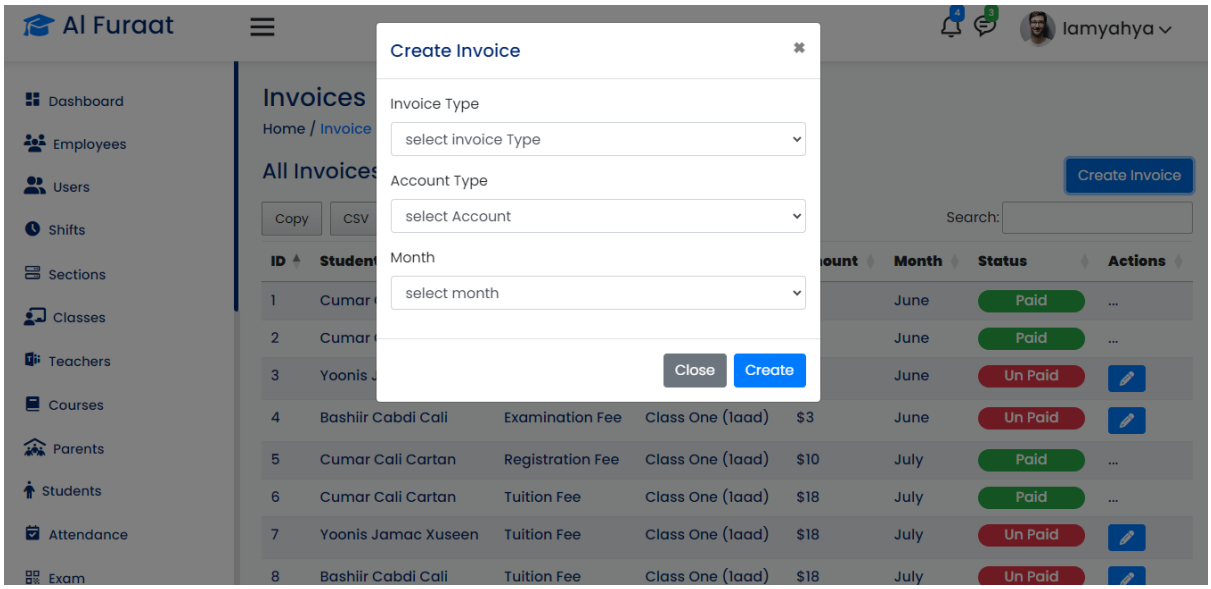


Figure 8.2.3 Finance

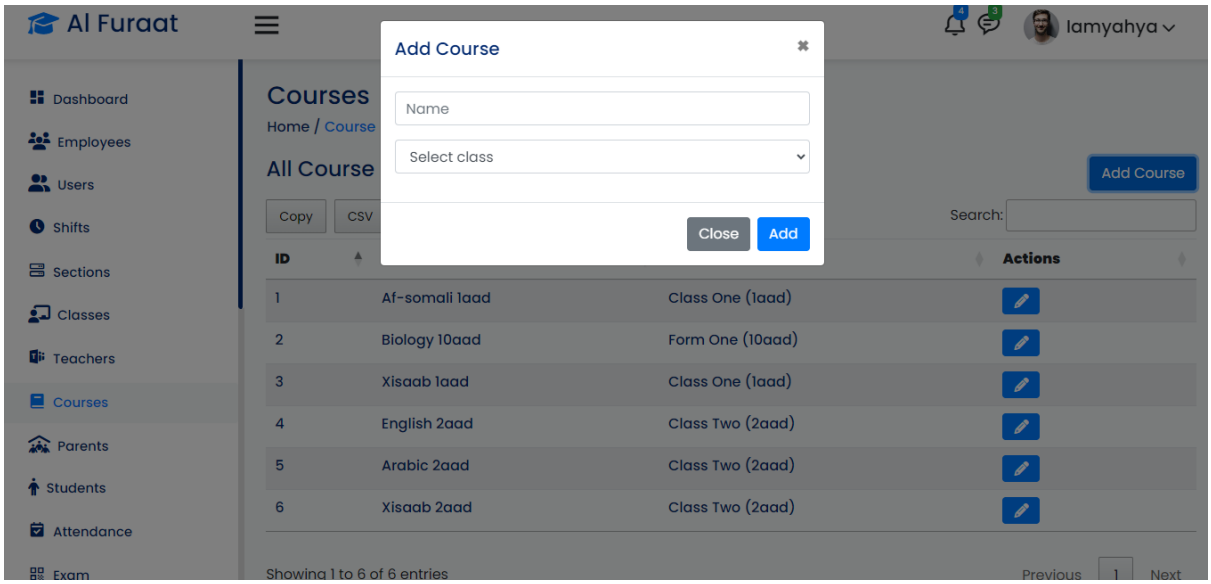


Figure 8.2.4 Add Course

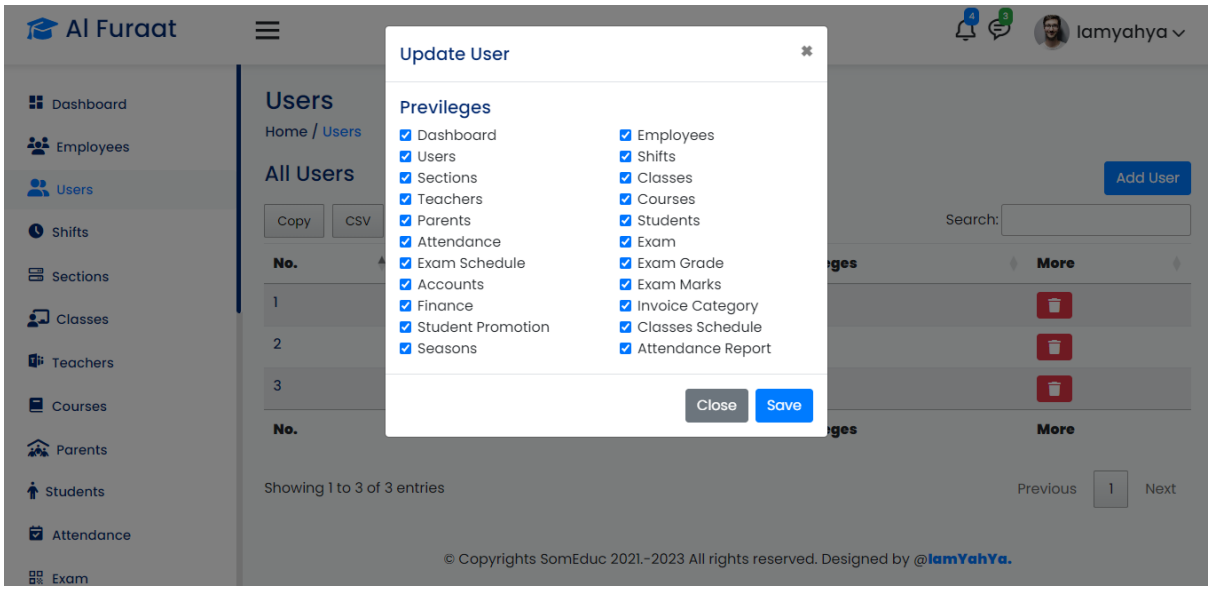


Figure 8.2.5 User Privileges

5.3 USER DOCUMENTATION

Documentation describes an information system and helps the users who must interact with it. Accurate documentation can reduce system downtime, cut cost, and speed up maintenance task.

Documentation is essential for successful system operation and maintenance. In

addition to supporting a system's users, accurate documentation is essential for developers who must modify the system, add new features or perform maintenance.

Documentation includes program documentation and user documentation.

5.3.1 Program Documentation

Program documentation describes the inputs, and processing logic for all program modules, the program documentation process starts in the system analysis phase and continues during systems implementation. In the system analysis phase will prepare overall documentation, such as process description and report layout, early in system development life cycle.

5.3.2 System Documentation

System documentation describes the system's functions and how they are implemented.

System documentation includes data dictionary entries, data flow diagrams, screen layout, source documents. System documentation is necessary reference material for Programmers who must support and maintain the system

5.3.3 User Documentation

User documentation consists of instructions and information to users who will interact with the system and includes user manuals help screen.

User documentation includes the following:-

- A system overview that clearly describes all major system features, capabilities, and limitations.
- Description of source document content, preparation, processing and samples.
- Overview of menu and data entry screen options, contents, and processing instruction.
- Examples of reports that are produced regularly or available at the users request, including samples.
- Explanation of responsibility for specific input, output, and processing requirements.
- Explanation of how to get help and procedures for updating user manual.

5.4 IMPLEMETATION

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, an evaluation of change over methods. 7 Apart from planning major task of preparing the implementation are education and training of users. Implementation is where implement how the system works as the sequential flow of Forms. As we mentioned above this application has different users and different privileges, so if you login the administrator you see the privileges and do everything in this system. While the others can see only their privileges and can register

5.5 CHPATER SUMMARY

Software testing means to uncover error before implementing systems. The three software testing strategies used are unit testing which focus to uncover errors within the boundary of the module. Integration testing a technique for constructing the architecture while at the same time conducting tests to uncover errors associated with interfacing of System testing.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.0 INTRODUCTION

Every IT project is bound to time, scope and goal. The need to upgrade systems and adding components never ends in developing systems because of changing business processes and business needs. So, to meet those business needs, it is better to conclude the IT projects with recommendations to precede future work.

6.1 CONCLUSIONS

Finally, this research work is attempted to make a student record system which reduces the manual way which has already used. The main objective of the system to develop an automated student record system to overcome all drawback consequences about customers and also minimize the consequences of using manual processing about the time taken to register customers.

As known, systems are developed using either structured analysis methodology or object oriented methodology. The UPMS was developed using the structured analysis that starts from systems development and ends in testing and implementing.

This is the last step of developing this project, we successfully completed other required steps, and normally every system has strength and weakness, so we want to mention them in the following lines:

6.1.1 Achievements

Every project has some strengths and weaknesses; so, we would like to identify some of the strengths and good features that student record management system will provide to the users. This system is very friendly system and it has a good interface that can be usable by every person who is computer literate. Moreover, if the user make a mistake it generate an error message that easily understandable by the user and it gives you the necessary utilities in you project, and other tools that you may need while you are working with the project. Our system has a capability to work 24hrs with addition and provides sufficiency and effective work and has equivalent storage.

6.1.2 Limitations

On the other hand, every project has its own Limitations, so, In this project, the only problems that you might face is alerting messages that appear whenever you misuse to the system, and if you are not familiar with this system you might find it difficult to manage it, please don't confuse, just read carefully what the message is carrying out and then click on the appropriate button. This system may only get problem towards the restriction format when using admin and user.

6.1.3 Future Enhancement

This project was developed by using different types of software, so, If some one tries to develop this project ,or enhance some of its functionalities and features , he/she should be able to know how to use PHP,MYSQL

6.2 RECOMMENDATIONS

Based on experience during the research, we would like to recommend that more research be done in student record system and a lot of literature review done for related work. In this study we focused on for student record systems which is as private student record management system.

REFERENCES

Basse, S., Okodoko, D., & Akpanumo, U. (2009). African Research Review. Information Communication Technologies in the Management of Education for Sustainable Development in Africa,3, 414-428.

Bunoti, S. (2004). The quality of higher education in developing countries needs professional support .

C.R, Kothari. (2004). Research methodology methods and techniques. (Second ed., p. 100). India: NEW AGE INTERNATIONAL (P). DOI: www.newagepublishers.com

Carcary M, Long G and Remenyi D (2007) "The Implementation of a New Student Management Information System (MIS) at an Irish Institute of Technology – An Ex Post Evaluation of its Success" The Electronic Journal Information Systems Evaluation Volume 10 Issue 1, pp 31 - 44, available online at www.ejise.com

Chifwep, V. (1995).Managing records at school level. National Education Statistical Information Systems.

Data Protection Principles in the Personal Data (Privacy) Ordinance from the Privacy Commissioner's perspective (2nd Edition) Website: www.pcpd.org.hk Enquiry Hotline: (852) 2827

Eudra, L. (2001). The rules governing medicinal products in the European union . Computerized Systems, 4

Gregor, S. (2002). A theory of theories in information systems. In S. Gregor and D. Hart (Eds.), Information Systems Foundations: Building the Theoretical Base. Australian National University, Canberra, 1-20.

Guidelines for Postsecondary Institutions for Implementation of the Family Educational Rights and Privacy Act of 1974 as Amended, American Association of Collegiate Registrars and Admissions Officers, Revised Edition 1995.

International Journall of Economy, Management and Social Sciences Vol (1), No (1), December 2012, pp. 1-18 http://www.waprogramming.com

Journal of Research Practice, 2(2), Article D1, (2006). Benfield, J. A., & Szlemko, W. J. (2006).

MIL-STD-498 (1994) Software Development and Documentation U.S department of Defense.

MoREQ (2001) Model Requirements for the Management of Electroni Records Cornwell Management Consultants http://www.cornwell.co.uk/moreq.html NA

Paul C. Hewett, Annabel S. Erulkar and Barbara S. Mensch (2003) the Feasibility of Computer-Assisted Survey Interviewing in Africa: Experience from Two Rural Districts in Kenya No.168

R., F. Lehner, M., & F., L. (2006). Perspectives on knowledge management systems theoretical framework and design of an empirical study. Management Information Systems III

Rope, M. (1999). Managing public sector record. United Kingdom: International Records Management Trust Retrieved from http://www.irmt.or

Shaw, A., & Shaw, D. (2006). Electronic Records Management Criteria and Information Security. Australian Information Warfare and Security Conference,7th. Retrieved March 19, 2015, from http://ro.ecu.edu.au/isw/

State of Florida General Records Schedule GS5 for University/Community College Records, March 1996.

Wokocha, C., & Adebayo, E. (2006). A Guide to Establishing Management Information System in Tertiary Institutions in Nigeria. European scientific journal,8(14), 12-12.

C. Y. (2014). Design and Implementation of Students' Information Management System. Student Information Management System, Microsoft Office Access, 103-107. Retrieved May 12, 2016.

APPENDIX A:

TIME FRAME WORK OF PROJECT

NO:	DATE	TASKS THAT THE RESEARCHER WILL DO FOR THIS STUDY	MONTH/WEEK/DAYS	REMARKS
1	1 /05/2023	Collection of literature review	1weeks	
2	14/05/2023	Literature review writing	2 week	
3	17/05/2023	Taking interview for the respondents	3 days	
4	19/05/2023	Collecting the research data	2 days	
6	20/05/2023	Interpreting the data and discussion	1 Day	
7	23/05/2023	Submission	3 days	
8	30/05/2023	Analysis Phase	1 weeks	
9	5/06/2023	Design phase	1 weeks	
10	20/06/2023	Program Coding	2 weeks	
11	5/07/2023	Testing and Implementation	1 weeks	
		TOTAL	**10 weeks**	

APPENDIX B:

BUDGET

No	Description	Cost
1	Internet Cost	$25
2	Printing & photocopy	$12
3	Electricity Cost	$8
4	Transportation Cost	$22
Total		$67