Bhai Mastan Singh Public School, Sri Muktsar Sahib

ACADEMIC YEAR : 2021-22

PROJECT REPORT ON

STUDENT MANAGEMENT SYSTEM

ROLL NO   :

NAME   :   Harpreet Kaur, Jasmeen kaur,Armanpreet Kaur

CLASS   :   XII

SUBJECT   :   Information Practices

SUB CODE   :   065

PROJECT GUIDE: Mrs.Monika

           PGT (IP)

        Bhai Mastan Singh Public School,
            Sri Muktsar Sahib

Bhai Mastan Singh Public School, Sri Muktsar Sahib

CERTIFICATE

This is to certify that Cadet _____

Roll No: _____ has successfully

completed the project Work entitled STUDENT MANAGEMENT SYSTEM

in the subject Information Practices (065) laid down in the

regulations of CBSE for the purpose   of   Practical

Examination in   Class XII to be held in Bhai Mastan Singh

Public School, Sri Muktsar Sahib on_____.

External:

Name: _____

Signature:

(Mrs.Monika)
PGT {IP}

Principal

TABLE OF CONTENTS [T O C]

ACKNOWLEDGEMENT

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I express deep sense of gratitude to almighty God for giving me strength for the successful completion of the project.

I express my heartfelt gratitude to my parents for constant encouragement while carrying out this project.

I gratefully acknowledge the contribution of the individuals who contributed in bringing this project up to this level, who continues to look after me despite my flaws,

I express my deep sense of gratitude to the luminary The Principal, Bhai Mastan Singh Pub.School who has been continuously motivating and extending their helping hand to us.

I am overwhelmed to express my thanks to The Administrative Officer for providing me an infrastructure and moral support while carrying out this project in the school.

My sincere thanks to   **Mrs.Monika**, Master In-charge, A guide, Mentor all the above a friend, who critically reviewed my project and helped in solving each and every problem, occurred during implementation of the project

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

<div align="center">

PROJECT ON STUDENT MANAGEMENT SYSTEM

</div>

INTRODUCTION

The project **Student Management System** is developed with a view to assist schools in their functioning. This project is aimed at providing easy access to the administration about their student records and manage their grades and report cards.

OBJECTIVES OF THE PROJECT

The objective of this project is to let the students apply the programming knowledge into a real- world situation/problem and exposed the students how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.
2. Apply object oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
5. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

PROPOSED SYSTEM

Today one cannot afford to rely on the fallible human beings of be really wants to stand against today's merciless

competition where not to wise saying **"to err is human"** no longer valid, it's outdated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of flies with a much sophisticated hard disk of the computer.

One has to use the data management software. Software has been an ascent in atomization various organisations. Many software products working are now in markets, which have helped in making the organizations work easier and efficiently. Data management initially had to maintain a lot of ledgers and a lot of paper work has to be done but now software product on this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done.

This prevents a lot of time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it's an age of computers of and automating such an organization gives the better look.

SYSTEM DEVELOPMENT LIFE CYCLE (*SDLC*)
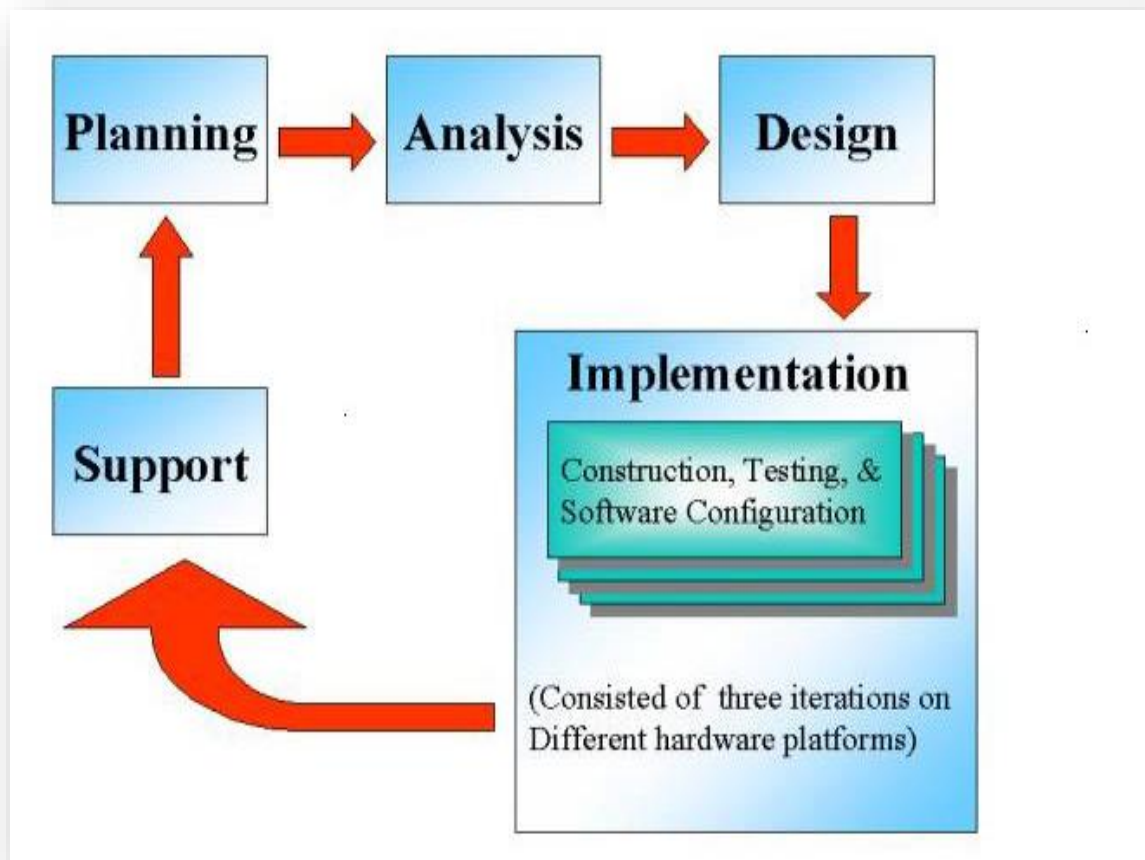
The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting projects allows managers to verify the successful completion of project phases before allocating resources to subsequent phases.

Software development projects typically include initiation, planning, design, development, testing, implementation, and maintenance phases. However, the phases may be divided differently depending on the organization involved.

For example, initial project activities might be designated as request, requirements-definition, and planning phases, or initiation, concept-development, and planning phases. End users of the system under development should be involved in reviewing the output of each phase to ensure the system is being built to deliver the needed functionality.

PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE

INITIATION PHASE

The Initiation Phase begins when a business sponsor identifies a need or an opportunity.

The purpose of the Initiation Phase is to:

- Identify and validate an opportunity to improve business accomplishments of the organization or a deficiency related to a business need.
- Identify significant assumptions and constraints on solutions to that need.
- Recommend the exploration of alternative concepts and methods to satisfy the need including questioning the need for technology, i.e., will a change in the business process offer a solution?
- Assure executive business and executive technical sponsorship. The Sponsor designates a Project Manager and the business need is documented in a Concept Proposal. The Concept Proposal includes information about the business process and the relationship to the Agency/Organization.
- Infrastructure and the Strategic Plan. A successful Concept Proposal results in a Project Management Charter which outlines the authority of the project manager to begin

the project.

Careful oversight is required to ensure projects support strategic business objectives and resources are effectively implemented into an organization's enterprise architecture. The initiation phase begins when an opportunity to add, improve, or correct a system is identified and formally requested through the presentation of a business case. The business case should, at a minimum, describe a proposal's purpose, identify expected benefits, and explain how the proposed system supports one of the organization's business strategies. The business case should also identify alternative solutions and detail as many informational, functional, and network requirements as possible.

SYSTEM CONCEPT DEVELOPMENT PHASE

The System Concept Development Phase begins after a business need or opportunity is validated by the Agency/Organization Program Leadership and the Agency/Organization CIO.

The purpose of the System Concept Development Phase is to:

- Determine the feasibility and appropriateness of the alternatives.
- Identify system interfaces.
- Identify basic functional and data requirements to satisfy the business need.
- Establish system boundaries; identify goals, objectives, critical success factors, and performance measures.
- Evaluate costs and benefits of alternative approaches to satisfy the basic functional requirements
- Assess project risks
- Identify and initiate risk mitigation actions, and Develop high-level technical architecture, process

models, data models, and a concept of operations. This phase explores potential technical solutions within the context of the business need.

- It may include several trade-off decisions such as the decision to use COTS software products as opposed to developing custom software or reusing software components, or the decision to use an incremental delivery versus a complete, onetime deployment.

- Construction of executable prototypes is encouraged to evaluate technology to support the business process. The System Boundary Document serves as an important reference document to support the Information Technology Project Request (ITPR) process.

- The ITPR must be approved by the State CIO before the project can move forward.

PICTORIAL REPRESENTATION OF SDLC:
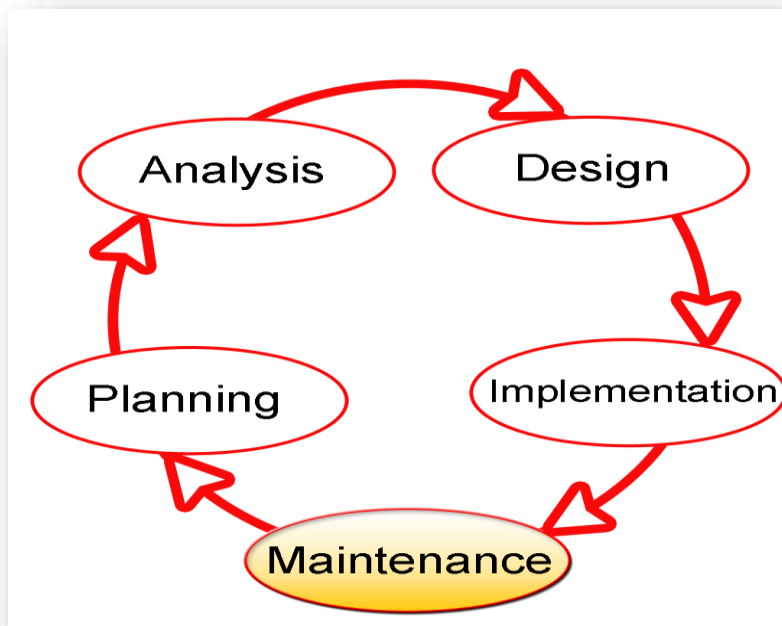
PLANNING PHASE

The planning phase is the most critical step in completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, is necessary to coordinate activities and manage project risks effectively. The depth and formality of project plans should be commensurate with the characteristics and risks of a given project. Project plans refine the information gathered during the initiation phase by further identifying the specific activities and resources required to complete a project.

A critical part of a project manager' sjob is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and network requirements as possible. During this phase, a plan is developed that documents the approach to be used and includes a discussion of methods, tools, tasks, resources, project schedules, and user input. Personnel assignments, costs, project schedule, and target dates are established.

A Project Management Plan is created with components related to acquisition planning, configuration management planning, quality assurance planning, concept of operations, system security, verification and validation, and systems engineering management planning.

REQUIREMENTS ANALYSIS PHASE

This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation, System Concept, and Planning phases. It also delineates the requirements in terms of data, system performance, security, and maintainability requirements for the system. The requirements are defined in this phase to a level of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the business

need or opportunity identified in the Initiation Phase. The requirements that will be used to determine acceptance of the system are captured in the Test and Evaluation Master Plan.

The purposes of this phase are to:

- Further define and refine the functional and data requirements and document them in the Requirements Document,
- Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated, who generates it, where does the information go, and who processes it),
- Develop detailed data and process models (system inputs, outputs, and the process.
- Develop the test and evaluation requirements that will be used to determine acceptable system performance.

DESIGN PHASE

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs are constructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections. Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database

layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design. Audit, security, and quality assurance personnel should be involved in the review and approval process. During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase could be very expensive to solve in the later stage of the software development, a variety of elements are considered in the design to mitigate risk. These include:

- Identifying potential risks and defining mitigating design features.
- Performing a security risk assessment.
- Developing a conversion plan to migrate current data to the new system.
- Determining the operating environment.
- Defining major subsystems and their inputs and outputs.
- Allocating processes to resources.
- Preparing detailed logic specifications for each software module. The result is a draft System Design Document which captures the preliminary design for the system.
- Everything requiring user input or approval is documented and reviewed by the user. Once these documents have been approved by the Agency CIO and Business Sponsor, the final System Design Document is created to serve as the Critical/Detailed Design for the system.
- This document receives a rigorous review by Agency technical and functional representatives to ensure that it satisfies the business requirements. Concurrent with the development of the system design, the Agency Project Manager begins development of the Implementation Plan, Operations and Maintenance Manual, and the Training Plan.

DEVELOPMENT PHASE

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. The procedures help ensure programmers clearly understand program designs and functional requirements. Programmers use various techniques to develop computer programs. The large transaction oriented programs associated with financial institutions have traditionally been developed using procedural programming techniques. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program. Effective completion of the previous stages is a key factor in the success of the Development phase. The Development phase consists of:

- Translating the detailed requirements and design into system components.
- Testing individual elements (units) for usability.
- Preparing for integration and testing of the IT system.

INTEGRATION AND TEST PHASE

- Subsystem integration, system, security, and user acceptance testing is conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. OIT Security staff assess the system security and issue a security certification and accreditation prior to installation/implementation.

Multiple levels of testing are performed, including:

- Testing at the development facility by the contractor and possibly supported by end users

- Testing as a deployed system with end users working together with contract personnel

- Operational testing by the end user alone performing all functions. Requirements are traced throughout testing, a final Independent Verification & Validation evaluation is performed and all documentation is reviewed and accepted prior to acceptance of the system.

IMPLEMENTATION PHASE

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

OPERATIONS AND MAINTENANCE PHASE

The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to the organization's needs. When modifications or
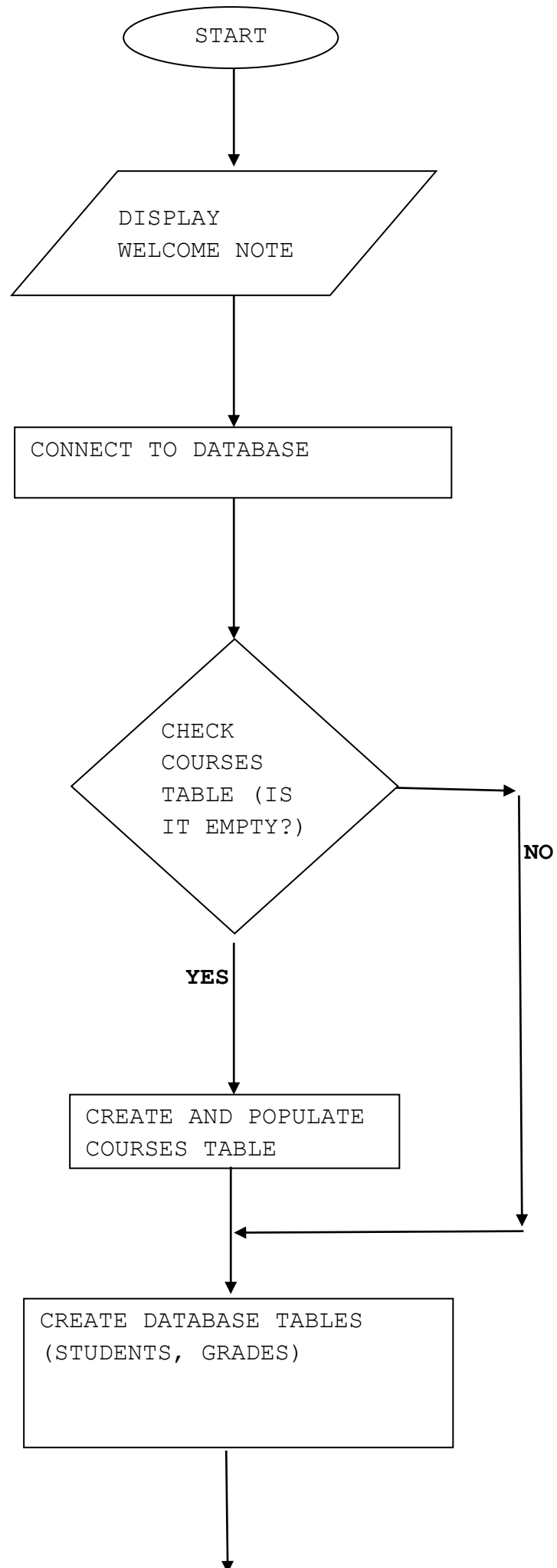
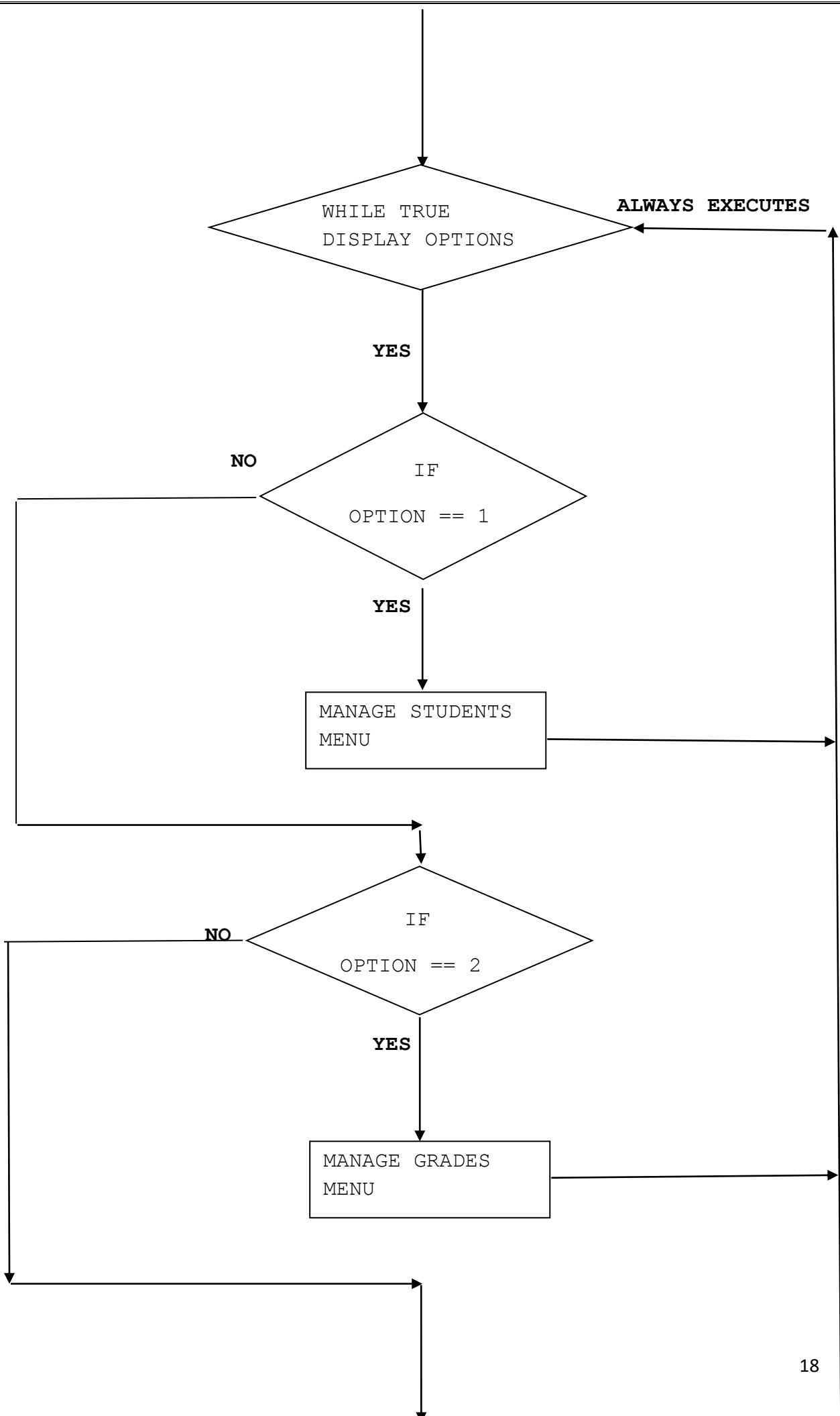changes are identified, the system may re-enter the planning phase.
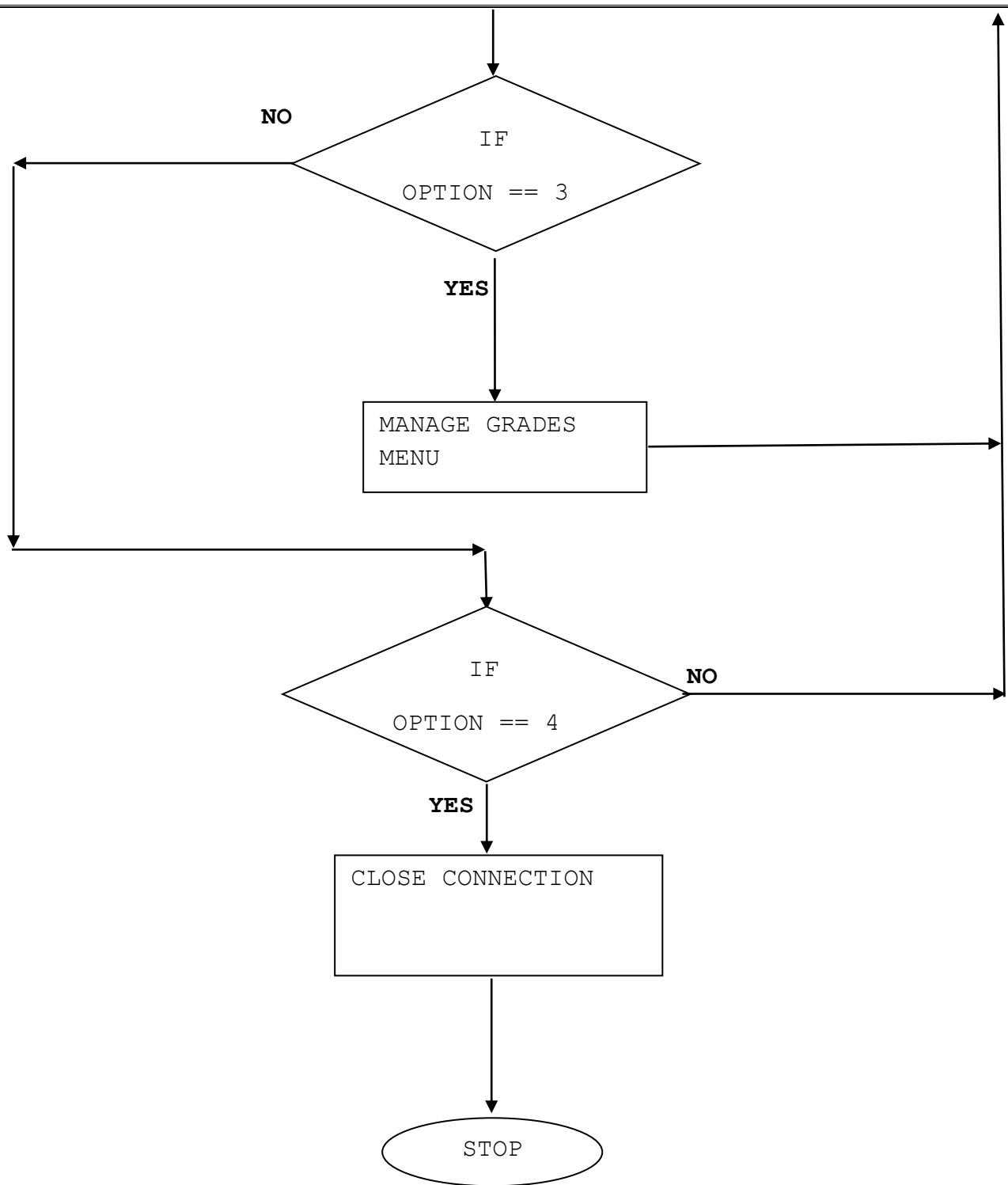
The purpose of this phase is to:

- Operate, maintain, and enhance the system.
- Certify that the system can process sensitive information.
- Conduct periodic assessments of the system to ensure the functional requirements continue to be satisfied.
- Determine when the system needs to be modernized, replaced, or retired.

FLOW CHART

```
                          ┌──────────┐
                         (   START    )
                          └──────────┘
                               │
                               ▼
                      ╱─────────────────╲
                     ╱   DISPLAY          ╲
                     ╲   WELCOME NOTE      ╱
                      ╲─────────────────╱
                               │
                               ▼
                    ┌─────────────────────────┐
                    │ CONNECT TO DATABASE      │
                    └─────────────────────────┘
                               │
                               ▼
                          ◇─────────◇
                         ╱  CHECK     ╲
                        ╱  COURSES     ╲──────────────► NO
                        ╲  TABLE (IS   ╱               │
                         ╲ IT EMPTY?) ╱                │
                          ◇─────────◇                 │
                               │                       │
                              YES                      │
                               │                       │
                               ▼                       │
                    ┌──────────────────────┐           │
                    │ CREATE AND POPULATE   │           │
                    │ COURSES TABLE         │           │
                    └──────────────────────┘           │
                               │                       │
                               ▼◄──────────────────────┘
                    ┌──────────────────────────┐
                    │ CREATE DATABASE TABLES    │
                    │ (STUDENTS, GRADES)        │
                    └──────────────────────────┘
                               │
                               ▼
```

17

WHILE TRUE
DISPLAY OPTIONS

ALWAYS EXECUTES

YES

IF
OPTION == 1

NO

YES

MANAGE STUDENTS
MENU

IF
OPTION == 2

NO

YES

MANAGE GRADES
MENU

18

```
                                          IF
              NO                       OPTION == 3
                                          YES


                          MANAGE GRADES
                          MENU


                                          IF
                                       OPTION == 4         NO

                                          YES

                          CLOSE CONNECTION


                              STOP
```

```python
import mysql.connector
import pandas as pd
import matplotlib.pyplot as plt


# Function Definitions


def print_welcome_note():
    print("""


================================================================
=================
                WELCOME TO BUNNY'S STUDENT MANAGEMENT SYSTEM


================================================================
=================
    """)


def create_database_connection(password):
    return mysql.connector.connect(host="localhost",
user="root", passwd=password, database="stud_manage")


def check_courses(cursor, db):
    cursor.execute("SELECT * FROM COURSES")
    data = cursor.fetchall()

    if len(data) == 0:
        courses_table(cursor, db)
    else:
        pass


def courses_table(cursor, db):
    cursor.execute("INSERT INTO COURSES VALUES (201,\"Python
and C\")")
```

```python
    cursor.execute("INSERT INTO COURSES VALUES (208,\"Data
Structures and Algorithm \")")
    cursor.execute("INSERT INTO COURSES VALUES (215,\"Security
and Cyber Laws\")")
    cursor.execute("INSERT INTO COURSES VALUES (211,\"Discrete
Mathematics\")")
    cursor.execute("INSERT INTO COURSES VALUES (212,\"Software
Engineering\")")
    cursor.execute("INSERT INTO COURSES VALUES
(213,\"Professional Skills and Ethics\")")
    cursor.execute("INSERT INTO COURSES VALUES (214,\"Design
and Analysis of Algorithm\")")
    db.commit()


def create_tables(cursor):
    cursor.execute("CREATE TABLE IF NOT EXISTS students (id
INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), age INT)")
    cursor.execute("CREATE TABLE IF NOT EXISTS courses
(course_id INT PRIMARY KEY, course_name VARCHAR(255))")
    cursor.execute("CREATE TABLE IF NOT EXISTS grades
(student_id INT, course_id INT, grade INT)")


def add_student(cursor, student_data):
    cursor.execute("SELECT NAME, AGE FROM STUDENTS")
    data = cursor.fetchall()
    flag = 0
    for name, age in data:
        if name == student_data[0] and age == student_data[1]:
            flag = 1
            print("\nSTUDENT ALREADY EXIST.")
    if flag == 0:
        cursor.execute("INSERT INTO students (name, age)
VALUES (%s, %s)", student_data)
        print("\nSTUDENT ADDED SUCCESSFULLY.")
```

```python
def update_student(cursor, student_id, updated_data):
    cursor.execute("SELECT ID FROM STUDENTS")
    data = cursor.fetchall()

    flag = 0
    for s_id in data:
        if str(s_id[0]) == student_id:
            flag = 1
            cursor.execute("UPDATE students SET name=%s,
age=%s WHERE id=%s", (updated_data['name'],
updated_data['age'], student_id))
            print("\nUPDATED SUCCESSFULLY.")
    if flag == 0:
        print("\nINVALID STUDENT ID.")


def delete_student(cursor, student_id):
    cursor.execute("SELECT ID FROM STUDENTS")
    data = cursor.fetchall()
    flag = 0
    for s_id in data:
        if str(s_id[0]) == student_id:
            flag = 1
            cursor.execute("DELETE FROM students WHERE id =
%s", (student_id,))
            print("\nDELETED SUCCESSFULLY.")
    if flag == 0:
        print("\nINVALID STUDENT ID.")


def get_student_data(cursor, student_id):
    cursor.execute("SELECT * FROM students WHERE id = %s",
(student_id,))
    data = cursor.fetchall()
    cols = ["ID", "NAME", "AGE"]
    df = pd.DataFrame(data, columns = cols)
    print("\n")
```

```python
        print(df)


    def list_all_students(cursor):
        cursor.execute("SELECT * FROM students")
        data = cursor.fetchall()
        cols = ["ID", "NAME", "AGE"]
        df = pd.DataFrame(data, columns = cols)
        print("\n")
        print(df)


        return cursor.fetchall()


    def assign_grade(cursor, student_id, course_id, grade):
        cursor.execute("SELECT STUDENT_ID, COURSE_ID FROM GRADES")
        data = cursor.fetchall()
        flag = 0
        for s_id, c_id in data:
            if str(s_id) == student_id and str(c_id) == course_id:
                flag = 1
                print("\nGRADES ALREADY ASSIGNED.")
                break
        if flag == 0:
            cursor.execute("INSERT INTO grades (student_id,
course_id, grade) VALUES (%s, %s, %s)", (student_id,
course_id, grade))
            print("\nGRADES ASSIGNED SUCCESSFULLY.")


    def update_grade(cursor, student_id, course_id, grade):
        cursor.execute("SELECT STUDENT_ID, COURSE_ID FROM GRADES")
        data = cursor.fetchall()
        flag = 0
        for s_id, c_id in data:
            if str(s_id) == student_id and str(c_id) == course_id:
                flag = 1
```

```python
                cursor.execute("UPDATE grades SET grade=%s WHERE
student_id=%s AND course_id=%s", (grade, student_id,
course_id))
            print("\nUPDATED SUCCESSFULLY.")
    if flag == 0:
        print("\nINVALID STUDENT OR COURSE ID.")


def generate_report(cursor):
    s_id = int(input("ENTER STUDENT ID:"))
    cursor.execute("SELECT ID, NAME FROM STUDENTS")
    stu_ids = cursor.fetchall()
    flag = 0
    for stu_id in stu_ids:
        if stu_id[0] == s_id:
            flag = 1
            cursor.execute(f"SELECT C.COURSE_NAME, G.GRADE
from COURSES C, GRADES G WHERE C.COURSE_ID = G.COURSE_ID AND
G.STUDENT_ID = '{s_id}'")
            data = cursor.fetchall()

            if len(data) == 0:
                print("\nSORRY... NO DATA.")
            else:
                df = pd.DataFrame(data, columns=['Course',
'Grade'])
                print("\n\t\t" + stu_id[1].upper() + "\'s
REPORT")
                print(df)
                df.plot(kind='bar', x='Course', y='Grade')
                plt.title(stu_id[1].upper())
                plt.ylabel("GRADES")
                plt.xlabel("COURSES")
                plt.xticks(rotation=9, fontsize = 8)
                plt.show()
```

```python
        if flag == 0:
            print("\nINVALID STUDENT ID.")



def print_main_menu():
    print("\nMain Menu")
    print("1. Manage Students")
    print("2. Manage Grades")
    print("3. Generate Reports")
    print("4. Exit")



def manage_students(cursor):
    while True:
        print("\n--- Manage Students ---")
        print("1. Add Student")
        print("2. Update Student")
        print("3. Delete Student")
        print("4. View Student Details")
        print("5. List All Students")
        print("6. Back to Main Menu")

        choice = input("Enter your choice: ")
        if choice == '1':
            name = input("Enter student name: ")
            age = int(input("Enter student age: "))
            add_student(cursor, (name, age))
        elif choice == '2':
            student_id = input("Enter student ID to update: ")
            name = input("Enter new name: ")
            age = input("Enter new age: ")
            update_student(cursor, student_id, {'name': name,
'age': age})
        elif choice == '3':
            student_id = input("Enter student ID to delete: ")
            delete_student(cursor, student_id)
```

```python
        elif choice == '4':
            student_id = input("Enter student ID: ")
            print("\nStudent Details:")
            get_student_data(cursor, student_id)

        elif choice == '5':
            students = list_all_students(cursor)
            for student in students:
                print(student)
        elif choice == '6':
            break

def manage_grades(cursor):
    while True:
        print("\n--- Manage Grades ---")
        print("1. Assign Grade")
        print("2. Update Grade")
        print("3. Back to Main Menu")

        choice = input("Enter your choice: ")
        cursor.execute("SELECT * FROM COURSES")
        data = cursor.fetchall()
        cols = ["COURSE ID", "COURSES"]
        df = pd.DataFrame(data, columns = cols)
        print("\n")
        print(df)
        if choice == '1':
            student_id = input("Enter student ID: ")
            course_id = input("Enter course ID: ")
            grade = input("Enter grade (OUT OF 10): ")
            assign_grade(cursor, student_id, course_id, grade)
        elif choice == '2':
            student_id = input("Enter student ID: ")
            course_id = input("Enter course ID: ")
            grade = input("Enter new grade: ")
```

```python
                    update_grade(cursor, student_id, course_id, grade)
                    print(f"Grade: {grade}")
            elif choice == '3':
                break


def main():
    db_connection = create_database_connection("root")  #
Replace with your actual password
    cursor = db_connection.cursor(buffered = True)

    create_tables(cursor)
    check_courses(cursor, db_connection)

    while True:
        print_main_menu()
        choice = input("\nEnter your choice: ")

        if choice == '1':
            manage_students(cursor)
        elif choice == '2':
            manage_grades(cursor)
        elif choice == '3':
            generate_report(cursor)
        elif choice == '4':
            print("Exiting the system. Goodbye!")
            break
        else:
            print("Invalid choice, please try again.")

        db_connection.commit()

    db_connection.close()


if __name__ == "__main__":
    main()
```

OUTPUT

File Edit Shell Debug Options Window Help

```
= RESTART: D:\12th projects\STUDENT MANAGEMENT\STUDENT MANAGEMENT SYSTEM FINAL.py

Main Menu
1. Manage Students
2. Manage Grades
3. Generate Reports
4. Exit

Enter your choice: 1

--- Manage Students ---
1. Add Student
2. Update Student
3. Delete Student
4. View Student Details
5. List All Students
6. Back to Main Menu
Enter your choice: 1
Enter student name: yash
Enter student age: 18

STUDENT ADDED SUCCESSFULLY.

--- Manage Students ---
1. Add Student
2. Update Student
3. Delete Student
4. View Student Details
5. List All Students
6. Back to Main Menu
Enter your choice: 4
Enter student ID: 23

Student Details:


Empty DataFrame
Columns: [ID, NAME, AGE]
Index: []
```

File Edit Shell Debug Options Window Help

```
--- Manage Students ---
1. Add Student
2. Update Student
3. Delete Student
4. View Student Details
5. List All Students
6. Back to Main Menu
Enter your choice: 5


   ID     NAME  AGE
0   6    bunny    6
1   8     ajay   61
2   9   monika   16
3  11  angelina  18
4  12    bunny    9
5  13    nigga   23
6  14     yash   18

--- Manage Students ---
1. Add Student
2. Update Student
3. Delete Student
4. View Student Details
5. List All Students
6. Back to Main Menu
Enter your choice: 2
Enter student ID to update: 9
Enter new name: ishant
Enter new age: 19

UPDATED SUCCESSFULLY.
```

28

```
--- Manage Students ---
1. Add Student
2. Update Student
3. Delete Student
4. View Student Details
5. List All Students
6. Back to Main Menu
Enter your choice: 3
Enter student ID to delete: 13

DELETED SUCCESSFULLY.

--- Manage Students ---
1. Add Student
2. Update Student
3. Delete Student
4. View Student Details
5. List All Students
6. Back to Main Menu
Enter your choice: 5


    ID    NAME  AGE
0   6     bunny   6
1   8      ajay  61
2   9    ishant  19
3  11  angelina  18
4  12     bunny   9
5  14      yash  18

--- Manage Students ---
1. Add Student
2. Update Student
3. Delete Student
4. View Student Details
5. List All Students
6. Back to Main Menu
Enter your choice: 4
Enter student ID: 12

Student Details:
```

```
--- Manage Students ---
1. Add Student
2. Update Student
3. Delete Student
4. View Student Details
5. List All Students
6. Back to Main Menu
Enter your choice: 4
Enter student ID: 12

Student Details:


    ID  NAME  AGE
0  12  bunny    9

--- Manage Students ---
1. Add Student
2. Update Student
3. Delete Student
4. View Student Details
5. List All Students
6. Back to Main Menu
Enter your choice: 6

Main Menu
1. Manage Students
2. Manage Grades
3. Generate Reports
4. Exit

Enter your choice: 2

--- Manage Grades ---
1. Assign Grade
2. Update Grade
3. Back to Main Menu
Enter your choice: 1
```

29

File Edit Shell Debug Options Window Help

```
--- Manage Grades ---
1. Assign Grade
2. Update Grade
3. Back to Main Menu
Enter your choice: 1


    COURSE ID                         COURSES
0       201                     Python and C
1       208     Data Structures and Algorithm
2       211              Discrete Mathematics
3       212              Software Engineering
4       213     Professional Skills and Ethics
5       214   Design and Analysis of Algorithm
6       215              Security and Cyber Laws
Enter student ID: 14
Enter course ID: 208
Enter grade (OUT OF 10): 9

GRADES ASSIGNED SUCCESSFULLY.

--- Manage Grades ---
1. Assign Grade
2. Update Grade
3. Back to Main Menu
Enter your choice: 2


    COURSE ID                         COURSES
0       201                     Python and C
1       208     Data Structures and Algorithm
2       211              Discrete Mathematics
3       212              Software Engineering
4       213     Professional Skills and Ethics
5       214   Design and Analysis of Algorithm
6       215              Security and Cyber Laws
Enter student ID: 12
Enter course ID: 211
Enter new grade: 3

INVALID STUDENT OR COURSE ID.
```

Ln: 247 Col: 0

19°C Sunny   ENG   14:48 27-12-2023

File Edit Shell Debug Options Window Help

```
--- Manage Grades ---
1. Assign Grade
2. Update Grade
3. Back to Main Menu
Enter your choice: 2


    COURSE ID                         COURSES
0       201                     Python and C
1       208     Data Structures and Algorithm
2       211              Discrete Mathematics
3       212              Software Engineering
4       213     Professional Skills and Ethics
5       214   Design and Analysis of Algorithm
6       215              Security and Cyber Laws
Enter student ID: 6
Enter course ID: 208
Enter new grade: 3

UPDATED SUCCESSFULLY.
Grade: 3

--- Manage Grades ---
1. Assign Grade
2. Update Grade
3. Back to Main Menu
Enter your choice: 3


    COURSE ID                         COURSES
0       201                     Python and C
1       208     Data Structures and Algorithm
2       211              Discrete Mathematics
3       212              Software Engineering
4       213     Professional Skills and Ethics
5       214   Design and Analysis of Algorithm
6       215              Security and Cyber Laws

Main Menu
1. Manage Students
2. Manage Grades
```
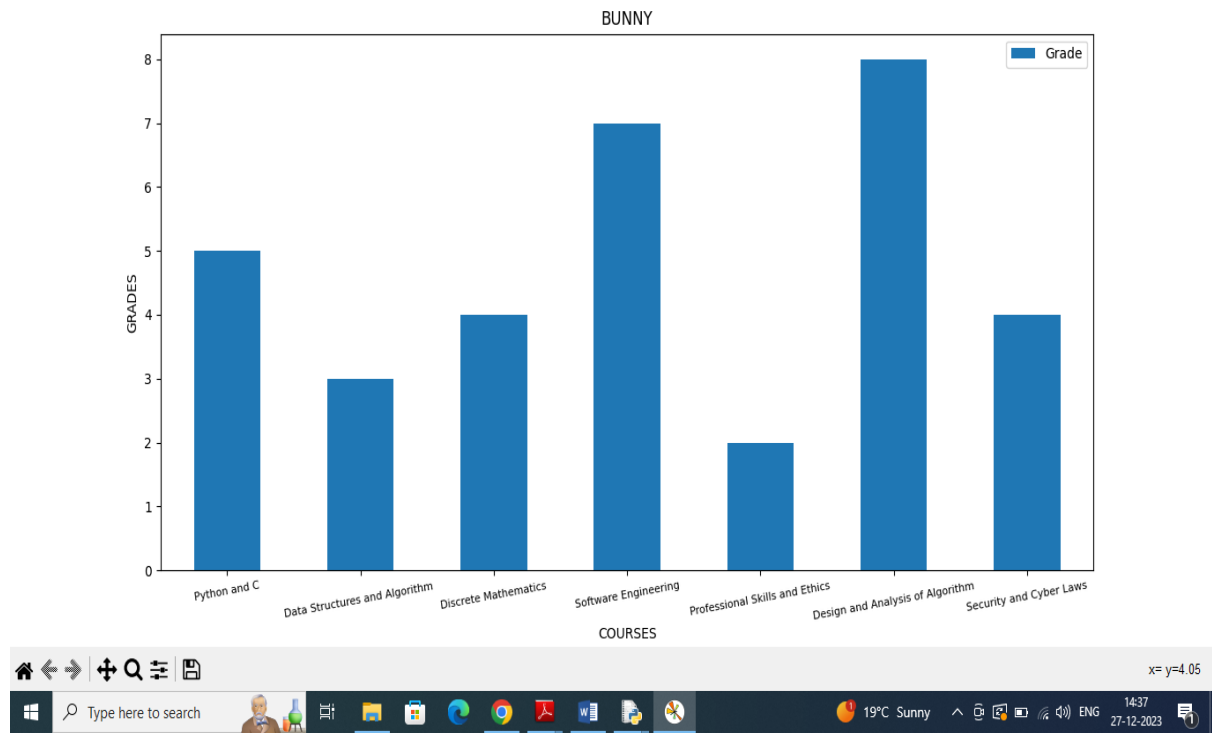
Ln: 247 Col: 0

19°C Sunny   ENG   14:49 27-12-2023

TESTING

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test[1] , with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics. Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

TESTING METHODS
Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

BLACK BOX TESTING
Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

SPECIFICATION-BASED TESTING

Specification-based testing aims to test the functionality of software according to the applicable requirements.[16] Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behaviour), either "is" or "is not" the same as the expected value specified in the test case. Specification-based testing is necessary, but it is insufficient to guard against certain risks

ADVANTAGES AND DISADVANTAGES

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code must have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers don't. But, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed.

That's why there are situations when (1) a black box tester writes many test cases to check something that can be tested by only one test case, and/or (2) some parts of the back end are not tested at all. Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

WHITE BOX TESTING

White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these)

Types of white box testing:-
The following types of white box testing exist:
- api testing - Testing of the application using Public and Private APIs.
- Code coverage - creating tests to satisfy some criteria of code coverage.

For example, the test designer can create tests to cause all statements in the program to be executed at least once.
- fault injection methods.
- mutation testing methods.
- static testing - White box testing includes all static testing.

CODE COMPLETENESS EVALUATION

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:
- Function Coverage: Which reports on functions executed and
- Statement Coverage: Which reports on the number of lines executed to complete the test.

They both return coverage metric, measured as a percentage

INSTALATION PROCEDURE

Student Management System :-

Pre-Requisites :-

1. You have to have the following softwares for the successful running of this software; which are

I) Python (Only for the First time), it is downloadable from 'www.python.org'.

II) MySQL (Only for the First time), it is downloadable from 'www.mysql.org'.

Installation :-

1. There will be two folders namely 'Python Files' and 'EXE files'.

2. The folder 'Python Files' will contain the source code of the software in python language. If you are running the software by the 3rd step mentioned below you have to pre install the following modules :-

I) mysql.connector

II) matplotlib.

III) pandas

3. Open the files in any python editors and run it to start and work on the software.

4. The folder 'EXE files' will contain two files namely 'main.exe' and 'Tables_in_mysql.exe'.

5. First run the 'Tables_in_mysql.exe' to create the tables in MySQL.

6. Then run the file 'main.exe' to start and work on the software.

CAUTION :-

If you are running the software through running the python files or by running the .exe files ; first run the file named 'Tables_in_mysql'.

The .exe file will take a lot of time; so be PATIENT.

HARDWARE AND SOFTWARE REQUIREMENTS

```
I.OPERATING SYSTEM        :      WINDOWS 7 AND ABOVE
II. PROCESSOR             :      PENTIUM(ANY) OR AMD
                                 ATHALON(3800+- 4200+ DUAL CORE)
III. MOTHERBOARD          :      1.845 OR 915,995 FOR PENTIUM 0R
MSI

                                 K9MM-V VIA K8M800+8237R PLUS
                                 CHIPSET FOR AMD ATHALON
IV. RAM                   :      512MB+
V. Hard disk              :       SATA 40 GB OR ABOVE
VI. CD/DVD r/w multi drive combo:   (If back up required)
VII. FLOPPY DRIVE 1.44 MB  :      (If Backup required)
VIII. MONITOR 14.1 or 15 -17 inch
IX. Key board and mouse
X. Printer                :      (if print is required –
[Hard copy])
```

SOFTWARE REQUIREMENTS:

  I.   Windows OS
  II. Python

BIBLIOGRAPHY

1. *Computer science With Python - Class XI   By : Sumita Arora*
2. *A Project Report On Blood Bank Management System (BBMS)*

   By : Praveen M Jigajinni
3. *Website:* **https://www.w3resource.com**

***