6/19/2025

Integrated Application Development

Java Programming & Database Management Systems

STUDENT MANAGEMENT SYSTEM

INTERGRATED APPLICATION DEVELOPMENT PROJECT

1. **Project Overview**

- Student Management System
- The integrated system will be used to manage student records that are focused on Students, Courses, Enrollment, Faculty and Grades. The interface is used by system administrators.
- **GROUP 7**

LEADER: Khanya Maqaqa (System Development)

MEMBERS:

- **Brandon Petersen (System Support)**
- **Lucky Sithole (System Development)**
- **Lesego Mbambo (System Development)**
- **Dimakatso (System Development)**
- **Fikile Masilo (System Development)**
- **Teboho Sepanya (System Development)**
- **Sara Semenya (System Development) (Minimal Participation)**

Date: 20 May 2025

Minutes of a meeting (Group Formation)

Time: 11:00 am

ATTENDANTS

- Teboho Sepanya (System Development)
- Brandon Petersen (System Support)
- Lucky Sithole (System Development)
- Lesego Mbambo (System Development)
- Dimakatso (System Development)
- Fikile Masilo (System Development)

Absenteeism

- Sara (System Development)

Meeting Adjourned

Time: 11:40 am

Date: 25 May 2025

Minutes of a meeting (Topic Selection Confirmation)

System Design / Documentation

- Sara (System Design) (Failed to Deliver)
- Brandon Petersen (Documentation) (System Design)

Database Design & Implementation

- Lesego Mbambo
- Dimakatso
- Lucky Sithole

Programming

- Khanya Maqaqa

System Testing & Deployment

Teboho Sepanya (Full Deployment) (Testing)

Brandon Petersen (Maintenance)

Fikile Masilo (Deployment) (Head Tester)

Absenteeism

Sara Semenya

Meeting Adjourn

Time: 11:25 am

Date: 10 June 2025

Minutes of a meeting (Progress Check)

Discussion

System Designers: Writing down the requirements of a system (e.g. ERDs, DFDs).

Documentation: Will document all contents and recordings gone throughout the project.

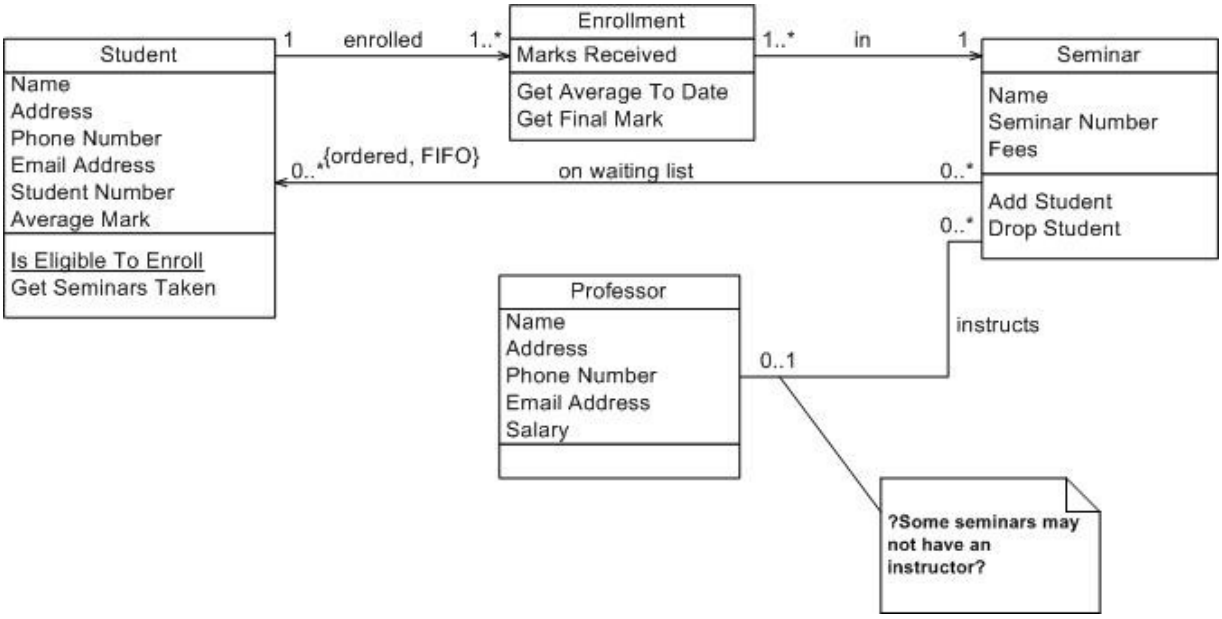
System Testing: Will be testing all system checks for errors. Recommendations were made to make improvements on student management systems and database.

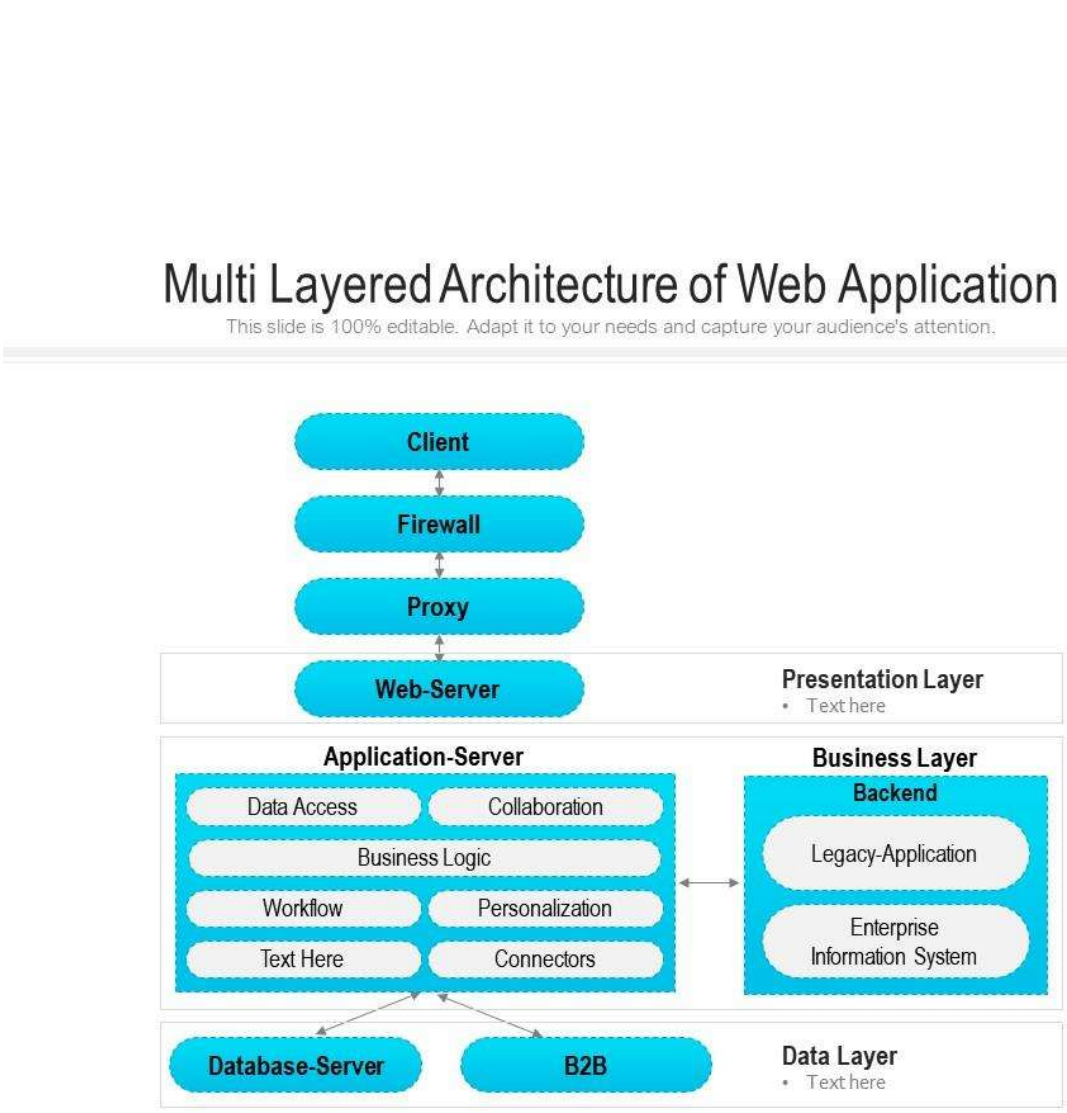
Deployment: Will give the project a go ahead.

Programming: Codes that will involve a text file.

Reports: Submitted from all departments

2. System Architecture

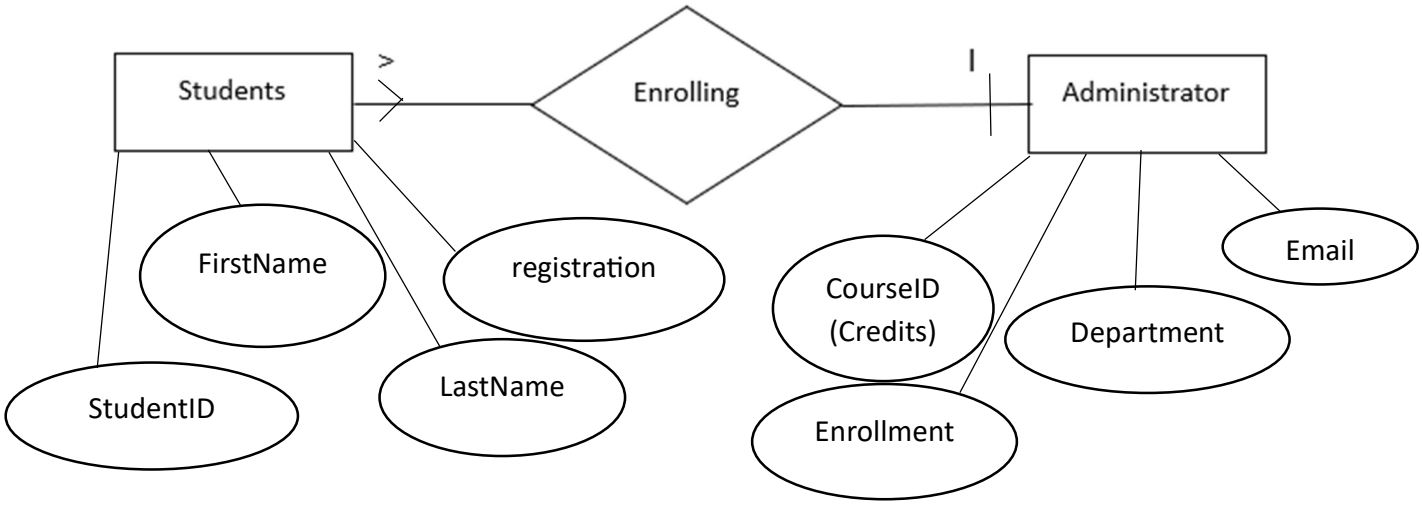




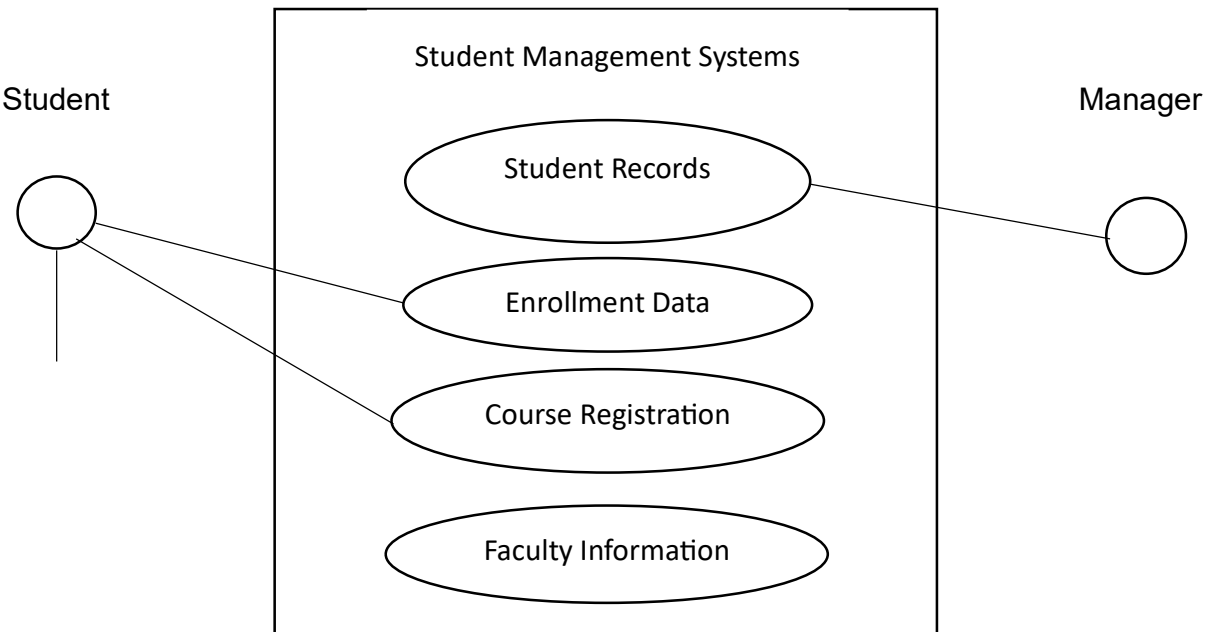
3. Database Design

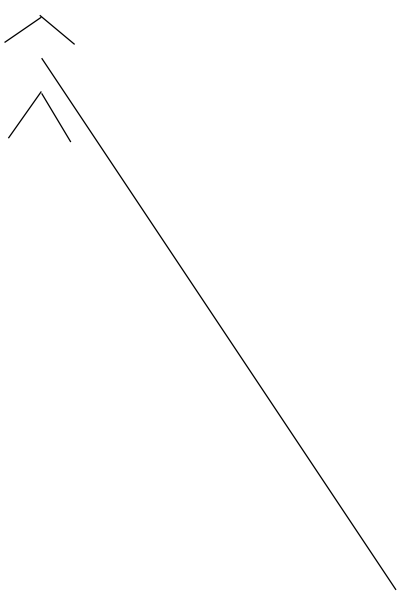
- UCD – Use Case Diagram
- DFD – Data Flow Diagram
- ERD – Student Management System
- List of Entities: Students, Administrators and Lecturers.
- Relationships: Studying, Enrolling and Learning.
- Database Normalization that uses a systematic process of organizing data to minimize redundancy and dependency.

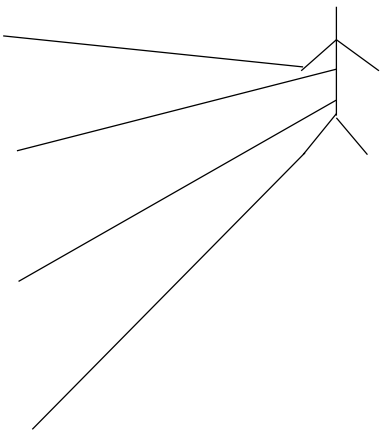
ERD – Entity Relationship Diagram



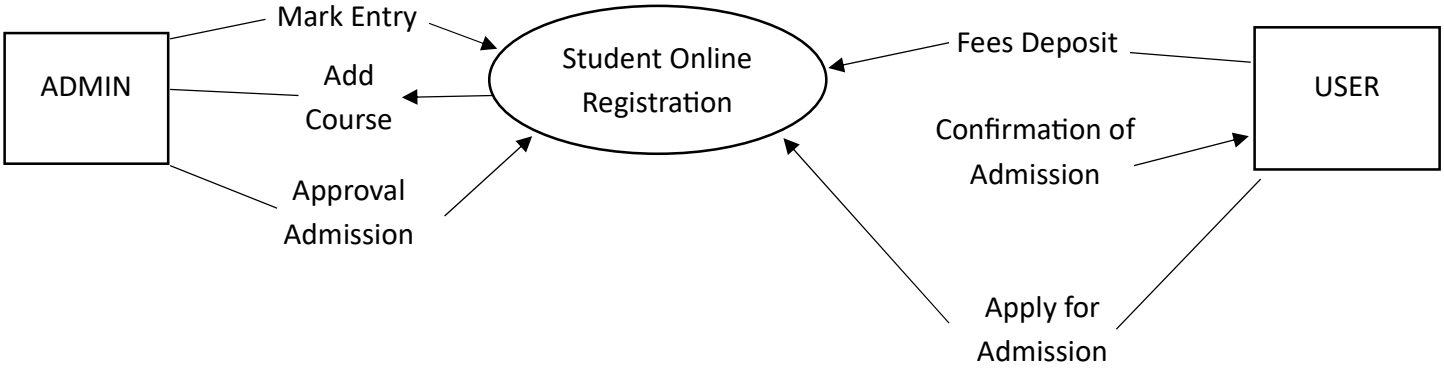
UCD – Use Case Diagram







Level 0 - DFD



- Actors: Entities interacting with the system
- Use Cases: Functional scenarios describing interactions
- Relationships: Association, Include, Extend.

Constraint Definitions:

Numeric Types:

• INT: Whole numbers (-2147483648 to 2147483647) e.g., 42, -1000, 8675309

• TINYINT: Small whole numbers (-128 to 127) e.g., 1, -99, 127

• SMALLINT: Medium whole numbers (-32768 to 32767) e.g., 5000, -20000

• MEDIUMINT: Larger whole numbers (-8388608 to 8388607) e.g., 1000000

• BIGINT: Very large whole numbers e.g., 9223372036854775807

• DECIMAL/NUMERIC: Precise decimal values e.g., 123.45, -56.789

• FLOAT/DOUBLE: Approximate decimal values e.g., 3.14159, -2.71828

• BIT: Boolean/binary values e.g., b'1', b'0', b'1010'

String Types:

• CHAR: Fixed-length strings e.g., 'A', 'abc', 'NY' (state code)

• VARCHAR: Variable-length strings e.g., 'Hello', 'user@example.com'

• BINARY/VARBINARY: Binary data e.g., file contents, encrypted data

• TEXT: Long text e.g., articles, comments, descriptions

• BLOB: Binary large objects e.g., images, documents, audio files

• ENUM: One of predefined values e.g., ('small', 'medium', 'large')

• SET: Multiple predefined values e.g., ('red', 'green', 'blue')

Date and Time Types:

• DATE: Calendar dates e.g., '2025-03-18'

• TIME: Time values e.g., '14:30:00'

• DATETIME: Combined date and time e.g., '2025-03-18 14:30:00'

• TIMESTAMP: Points in time e.g., '2025-03-18 14:30:00'

• YEAR: Year values e.g., 2025, 199

4. **Functionality Implementation**

- Java Code Development Platform Binary
- Visual Studio Code

KEY FEATURES IMPLEMENTED:

Steps taken to Use Scanner

The Scanner class in Java is used to get user input from the console. It's a powerful tool that allows your programs to interact with users by reading their input.

1. Import the Scanner class at the top of your program:

2. import java.util.Scanner;

3. Create a Scanner object:

4. Scanner s = new Scanner(System.in);

5. Use various methods to read different types of input:

- **`nextLine()` - reads a line of text (String)**
- **`nextInt()` - reads an integer**
- **`nextDouble()` - reads a decimal number**
- **`next()` - reads a single word**
- **`nextBoolean()` - reads a boolean value**

6. Close the Scanner when you're done:

7. `scanner.close();`

Here's a simple example that demonstrates how to use Scanner to get different types of input from the user:

Code:

```
package scannerexample;
```

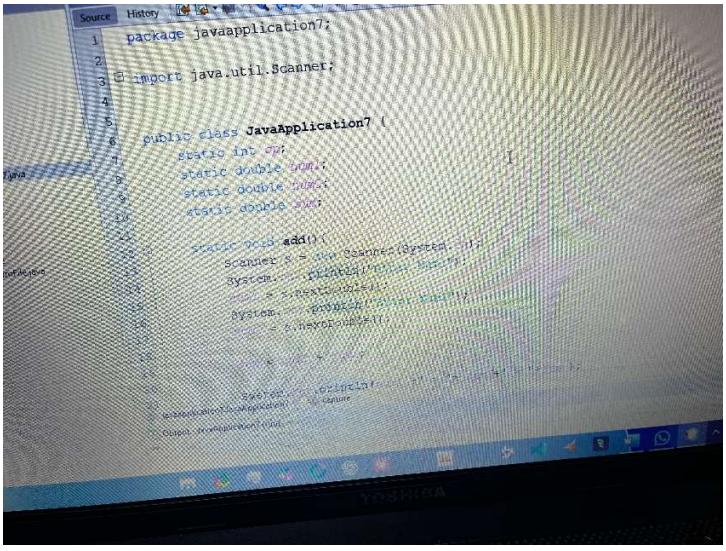
```
import java.util.Scanner;
public class ScannerExample {
        public static void main(String[] args) {
                // Create a Scanner object
                Scanner scanner = new Scanner(System.in);
                // Get the student's name
                System.out.print("Enter student name: ");
                String name = scanner.nextLine();
                // Get the student's age
               System.out.print("Enter student age: ");
                int age = scanner.nextInt();
                // Get the student's grade
                System.out.print("Enter student grade: ");
                double grade = scanner.nextDouble();
                // Display the information
                System.out.println("Student Information:");
                System.out.println("Name: " + name);
                System.out.println("Age: " + age);
                System.out.println("Grade: " + grade);
                // Close the Scanner
                scanner.close();
}
}
```

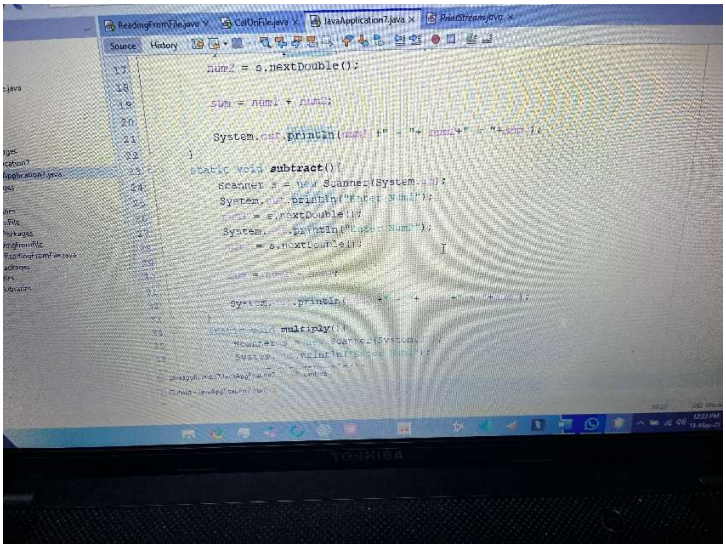
Expected Output

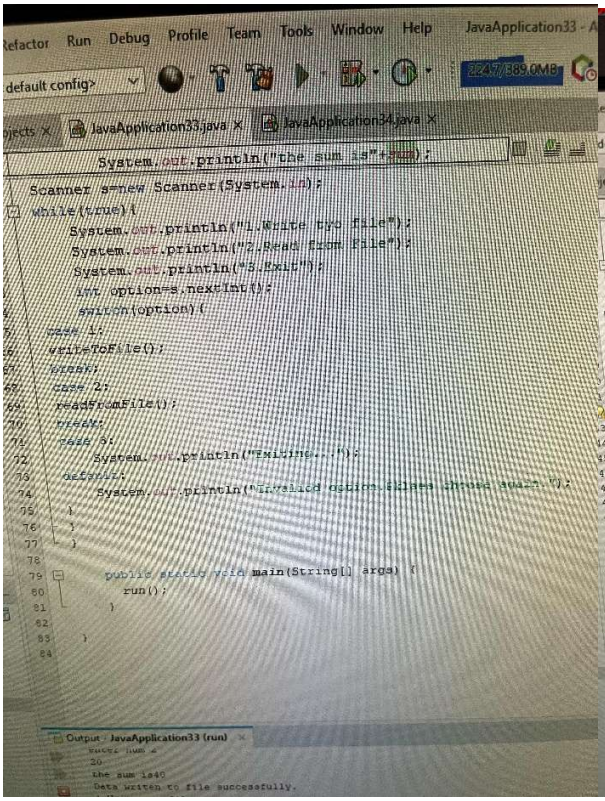
```
Enter student name: John Smith
```

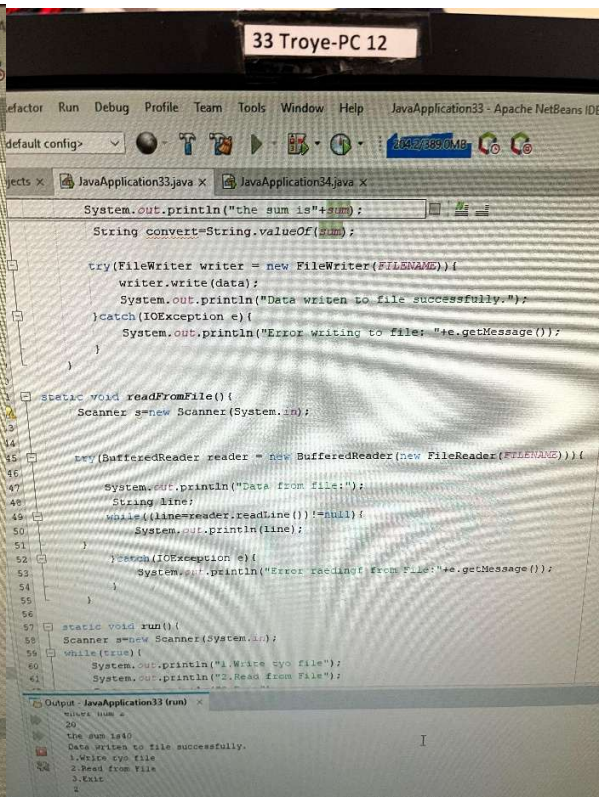
```
Enter student age: 18
Enter student grade: 85.5
Student Information:
Name: John Smith
Age: 18
Grade: 85.5
```

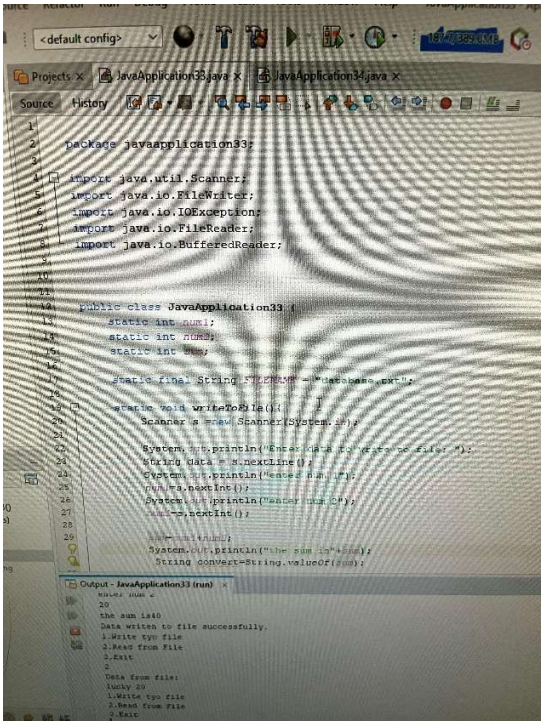
Java Code Development Platform Binary

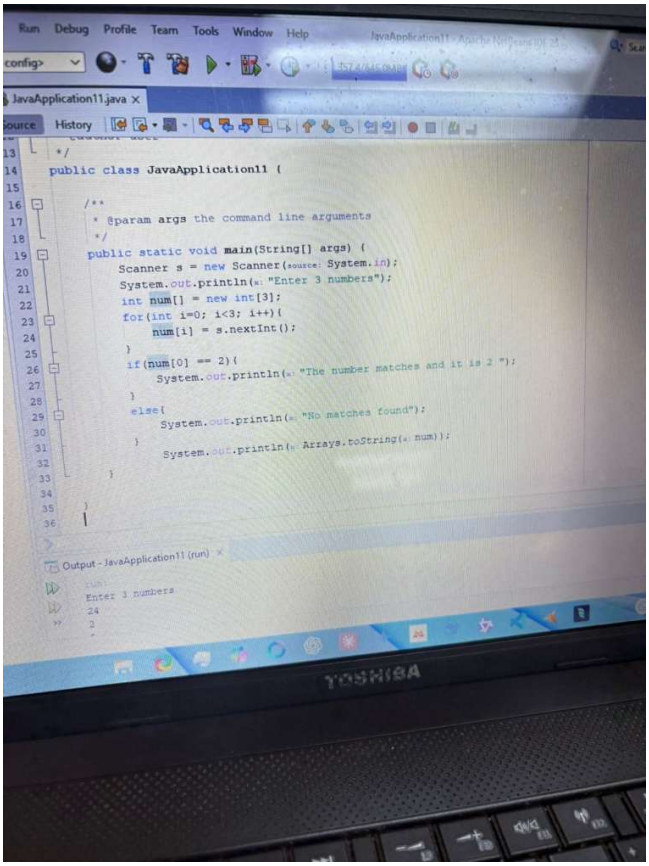

Source
History
package javaapplication7;
import java.util.Scanner;
public class JavaApplication7 {
static int op;
static double num1;
static double num2;
static double sum;
static void add()
Scanner s = new Scanner(System.in);
System.out.println("Enter Num1");
num1 = s.nextDouble();
System.out.println("Enter Num2");
num2 = s.nextDouble();
sum = num1 + num2;
System.out.println(num1+" + "+num2+" = "+sum);
}
static void subtract()
Scanner s = new Scanner(System.in);
System.out.println("Enter Num1");
num1 = s.nextDouble();
System.out.println("Enter Num2");
num2 = s.nextDouble();
sum = num1 - num2;
System.out.println(num1+" - "+num2+" = "+sum);
}
static void multiply()
Scanner s = new Scanner(System.in);
System.out.println("Enter Num1");
TOSHIBA

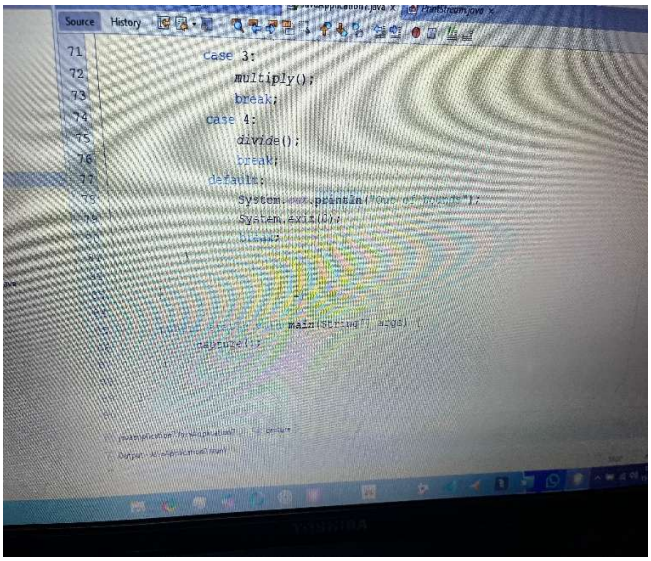

ReadingFromFile.java
CallOnFile.java
JavaApplication7.java
PrintStream.java
Source
History
num2 = s.nextDouble();
sum = num1 + num2;
System.out.println(num1+" + "+num2+" = "+sum);
}
static void subtract()
Scanner s = new Scanner(System.in);
System.out.println("Enter Num1");
num1 = s.nextDouble();
System.out.println("Enter Num2");
num2 = s.nextDouble();
sum = num1 - num2;
System.out.println(num1+" - "+num2+" = "+sum);
}
static void multiply()
Scanner s = new Scanner(System.in);
System.out.println("Enter Num1");
TOSHIBA

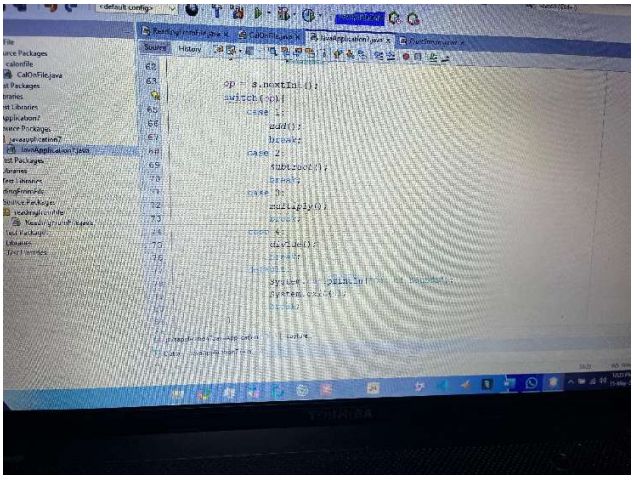


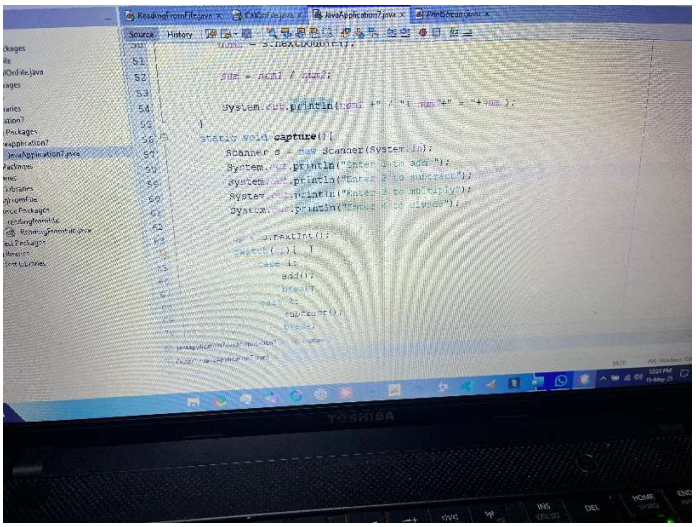

33 Troye-PC 12

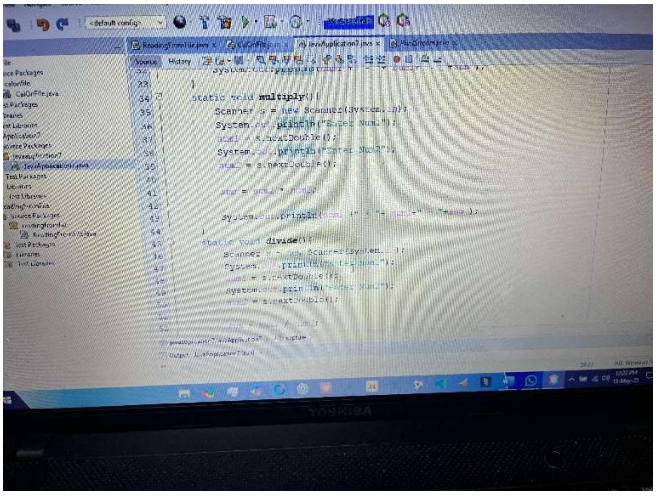



Run Debug Profile Team Tools Window Help
JavaApplication11.java
History
public class JavaApplication11 {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
Scanner s = new Scanner(source: System.in);
System.out.println(x: "Enter 3 numbers");
int num[] = new int[3];
for(int i=0; i<3; i++){
num[i] = s.nextInt();
}
if(num[0] == 2){
System.out.println(x: "The number matches and it is 2 ");
}
else{
System.out.println(x: "No matches found");
}
System.out.println(x: Arrays.toString(a: num));
}
}
Output - JavaApplication11 (run)
Enter 3 numbers
24
2
TOSHIBA


Source
History
case 3:
multiply();
break;
case 4:
divide();
break;
default:
System.exit(0);
main(String[] args)
TOSHIBA







JavaScript

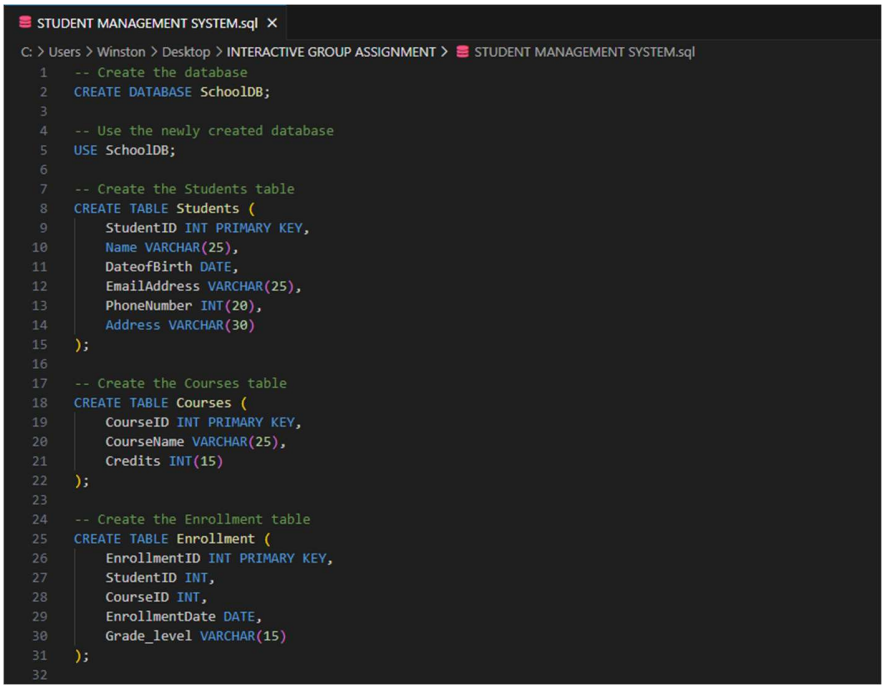
HTML to define the content of web pages

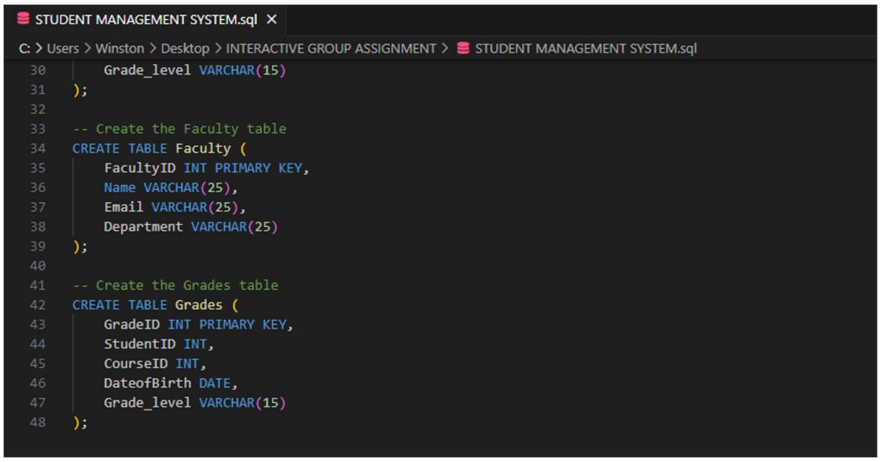
CSS to specify the layout of web pages

JavaScript to program the behaviour of web pages

SQL (Visual Studio Code)

Student Management System



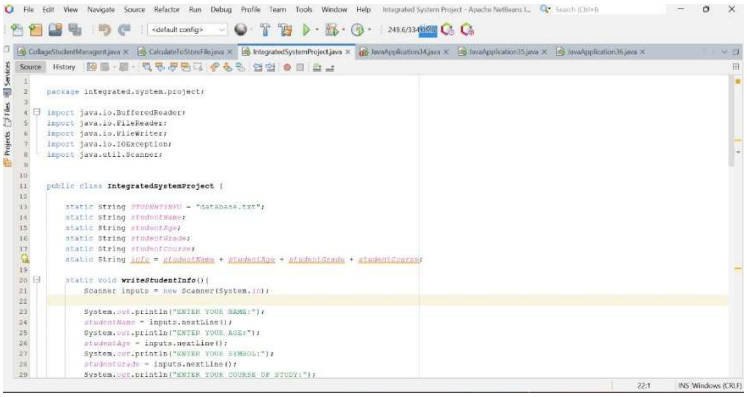


Important SQL Commands

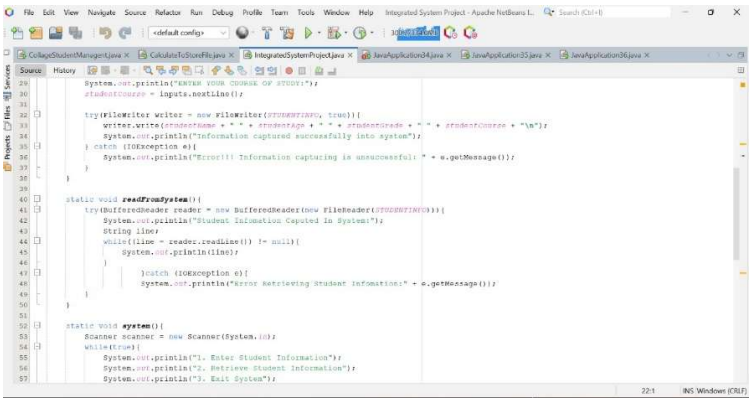
- SELECT – Extracts data from a database
- UPDATE – Updates data in a database
- DELETE – Deletes data from a database
- INSERT INTO – Inserts new data into a database
- CREATE DATABASE – Creates a new database
- ALTER DATABASE – Modifies a database
- CREATE TABLE – Creates a new table
- ALTER TABLE – Modifies a table
- DROP TABLE – Deletes a table
- CREATE INDEX – Creates an index (search key)
- DROP INDEX – Deletes an index

5. **Testing and Quality Assurance**

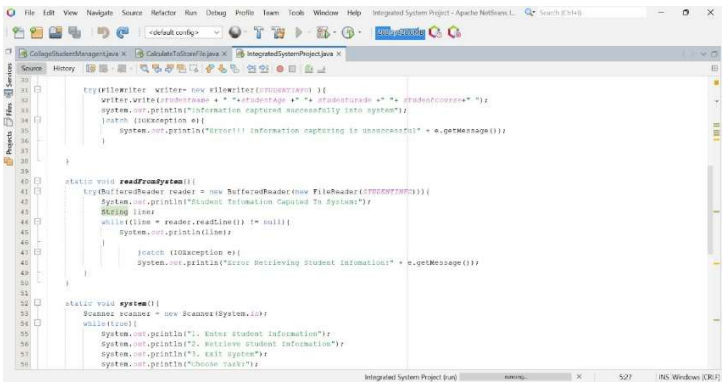
PROTOTYPE PHASE



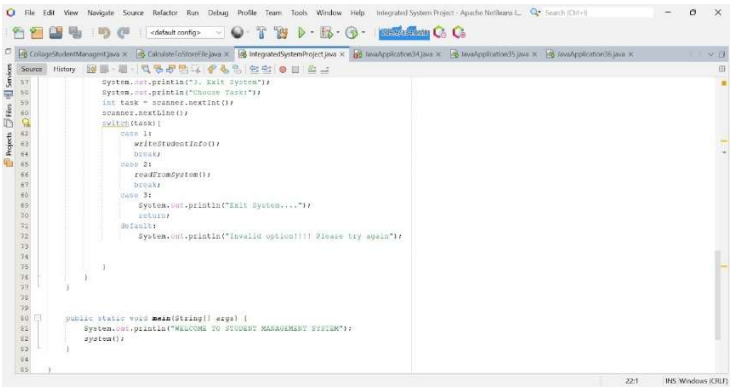
```
package integrated.system.project;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;


public class IntegratedSystemProject {

    static String STUDENTINFO = "database.txt";
    static String studentName;
    static String studentAge;
    static String studentGrade;
    static String studentCourse;
    static String info = studentName + studentAge + studentGrade + studentCourse;

    static void writeStudentInfo(){
        Scanner inputs = new Scanner(System.in);

        System.out.println("ENTER YOUR NAME:");
        studentName = inputs.nextLine();
        System.out.println("ENTER YOUR AGE:");
        studentAge = inputs.nextLine();
        System.out.println("ENTER YOUR SYMBOL:");
        studentGrade = inputs.nextLine();
        System.out.println("ENTER YOUR COURSE OF STUDY:");
```

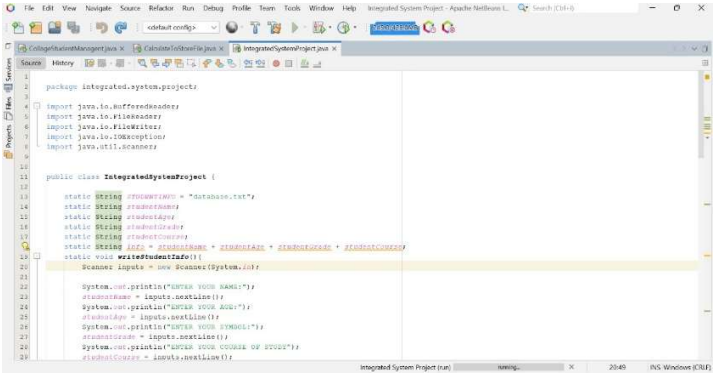


```
        System.out.println("ENTER YOUR COURSE OF STUDY:");
        studentCourse = inputs.nextLine();

        try(FileWriter writer = new FileWriter(STUDENTINFO, true)){
            writer.write(studentName + " " + studentAge + " " + studentGrade + " " + studentCourse + "\n");
            System.out.println("Information captured successfully into system");
        } catch (IOException e){
            System.out.println("Error!!! Information capturing is unsuccessful: " + e.getMessage());
        }
    }

    static void readFromSystem(){
        try(BufferedReader reader = new BufferedReader(new FileReader(STUDENTINFO))){
            System.out.println("Student Information Caputed In System:");
            String line;
            while((line = reader.readLine()) != null){
                System.out.println(line);
            }
                }catch (IOException e){
                System.out.println("Error Retrieving Student Infomation:" + e.getMessage());
        }
    }

    static void system(){
        Scanner scanner = new Scanner(System.in);
        while(true){
            System.out.println("1. Enter Student Information");
            System.out.println("2. Retrieve Student Information");
            System.out.println("3. Exit System");
```



```
        try(FileWriter  writer= new FileWriter(STUDENTINFO) ){
            writer.write(studentName + " "+studentAge +" "+ studentGrade +" "+ studentCourse+" ");
            System.out.println("Information captured successfully into system");
            }catch (IOException e){
                System.out.println("Error!!! Information capturing is unsuccessful" + e.getMessage());
            }

    }

    static void readFromSystem(){
        try(BufferedReader reader = new BufferedReader(new FileReader(STUDENTINFO))){
            System.out.println("Student Information Caputed In System:");
            String line;
            while((line = reader.readLine()) != null){
                System.out.println(line);
            }
                }catch (IOException e){
                System.out.println("Error Retrieving Student Infomation:" + e.getMessage());
        }
    }

    static void system(){
        Scanner scanner = new Scanner(System.in);
        while(true){
            System.out.println("1. Enter student information");
            System.out.println("2. Retrieve student information");
            System.out.println("3. exit system");
            System.out.println("Choose task:");
```

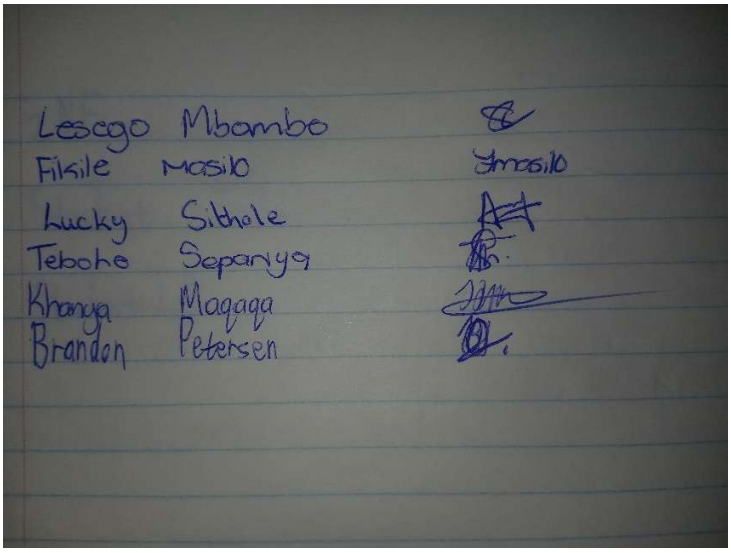




Contract disclosure

This document is signed under group members to verify its authentication and approval

SIGNATURES:



SYSTEM METHODOLOGY USED:

Waterfall Model

- *Requirement Gathering and Analysis*
- *System Design*
- *Implementation (Coding)*
- *Testing*
- *Deployment*
- *Maintenance*

AUTHOR OF DOCUMENTATION:

BRANDON WINSTON PETERSEN

TABLE OF CONTENTS:

2. System Architecture

• UML class diagram showing class relationships

• Description of each class's purpose and its relationship to database entities

• Diagram showing the system layers (presentation, business logic, data access)

3. Database Design

• Complete Entity-Relationship Diagram (ERD)

• Description of all entities and their relationships

• Explanation of normalization decisions

• Data dictionary with detailed field descriptions

• Constraint definitions and justifications

4. Functionality Implementation

• List and explanation of key features implemented

• Description of algorithms and data structures used

• Screenshots of important functionalities in action

• Explanation of key SQL operations and their role in the application

5. Testing and Quality Assurance

• Description of test cases and testing approach

• Bugs encountered and fixes implemented

• Limitations of the current implementation and future improvements

6. TERMS AND CONDITIONS

- **Contract Disclosure**

- **Signatures (Group Approval)**
- **System Methodology**
- **Author**

INTERGRATED SYSTEM PROJECT SCREENSHOTS

Integreted System Project.java

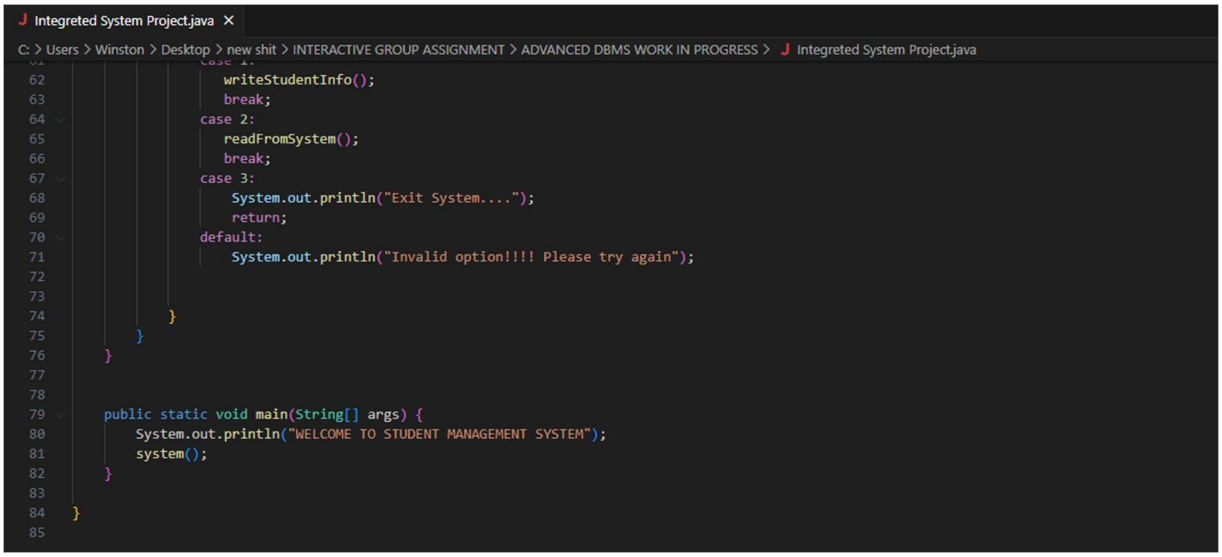
C: > Users > Winston > Desktop > new shit > INTERACTIVE GROUP ASSIGNMENT > ADVANCED DBMS WORK IN PROGRESS > Integreted System Project.java

```java
package integrated.system.project;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;


public class IntegratedSystemProject {

    static String STUDENTINFO = "database.txt";
    static String studentName;
    static String studentAge;
    static String studentGrade;
    static String studentCourse;
    static String info = studentName + studentAge + studentGrade + studentCourse;
    static void writeStudentInfo(){
        Scanner inputs = new Scanner(System.in);

        System.out.println("ENTER YOUR NAME:");
        studentName = inputs.nextLine();
        System.out.println("ENTER YOUR AGE:");
        studentAge = inputs.nextLine();
        System.out.println("ENTER YOUR SYMBOL:");
        studentGrade = inputs.nextLine();
        System.out.println("ENTER YOUR COURSE OF STUDY");
        studentCourse = inputs.nextLine();

        try(FileWriter  writer= new FileWriter(STUDENTINFO) ){
            writer.write(studentName + " "+studentAge +" "+ studentGrade +" "+ studentCourse+" ");
            System.out.println("Information captured successfully into system");
            }catch (IOException e){
```

Integreted System Project.java

C: > Users > Winston > Desktop > new shit > INTERACTIVE GROUP ASSIGNMENT > ADVANCED DBMS WORK IN PROGRESS > Integreted System Project.java

```java
            }catch (IOException e){
                System.out.println("Error!!! Information capturing is unsuccessful" + e.getMessage());
            }

    }

    static void readFromSystem(){
        try(BufferedReader reader = new BufferedReader(new FileReader(STUDENTINFO))){
            System.out.println("Student Infomation Caputed In System:");
            String line;
            while((line = reader.readLine()) != null){
                System.out.println(line);
            }
                }catch (IOException e){
                System.out.println("Error Retrieving Student Infomation:" + e.getMessage());
        }
    }

    static void system(){
        Scanner scanner = new Scanner(System.in);
        while(true){
            System.out.println("1. Enter Student Information");
            System.out.println("2. Retrieve Student Information");
            System.out.println("3. Exit System");
            System.out.println("Choose Task:");
            int task = scanner.nextInt();
            scanner.nextLine();
            switch(task){
                case 1:
                   writeStudentInfo();
                   break;
                case 2:
```


Integreted System Project.java
C: > Users > Winston > Desktop > new shit > INTERACTIVE GROUP ASSIGNMENT > ADVANCED DBMS WORK IN PROGRESS > Integreted System Project.java
writeStudentInfo();
break;
case 2:
readFromSystem();
break;
case 3:
System.out.println("Exit System....");
return;
default:
System.out.println("Invalid option!!!! Please try again");
public static void main(String[] args) {
System.out.println("WELCOME TO STUDENT MANAGEMENT SYSTEM");
system();

Final System

```
package integrated.system.project;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

public class IntegratedSystemProject {


    static String STUDENTINFO = "database.txt";

   static String studentSurname;

    static String UserName;

    static String password;

    static String studentName;

    static String department;

    static String dateOfBirth;

    static String emailAddress;

    static String phoneNumber;

    static String address;

    static String studentAge;

    static String studentGrade;

    static String studentCourse;

    static String enrollmentDate;

    static String info = studentName + studentSurname +studentAge + studentGrade + studentCourse + department
+dateOfBirth + emailAddress + phoneNumber+ address+ enrollmentDate;

    static void UserLogin(){

        Scanner s = new Scanner(System.in);

        System.out.println("Please Enter your credentials to Proceed");

        System.out.println("Enter Your UserName:");
```

```
    String enteredUser = s.nextLine();
    System.out.println("-------------------------------------");
    System.out.println("Enter Your Password:");
    String enteredPassword = s.nextLine();
    System.out.println("-------------------------------------");

    if(enteredUser.equalsIgnoreCase("Techsyn") && enteredPassword.equalsIgnoreCase("4231")){
        System.out.println("*****Login Successful*****");
        system();
    }
    else{
        System.out.println("Wrong details");
        UserLogin();

   }
 }

  static void writeStudentInfo(){
    Scanner inputs = new Scanner(System.in);
System.out.println("--------------------------");
    System.out.println("ENTER NAME:");
    System.out.println("USE CAPITAL LETTERS");
    studentName = inputs.nextLine();
    System.out.println("--------------------------");
    System.out.println("ENTER SURNAME");
    System.out.println("USE CAPITAL LETTERS");
    studentSurname = inputs.nextLine();
    System.out.println("--------------------------");
    System.out.println("ENTER AGE:");
    studentAge = inputs.nextLine();
    System.out.println("--------------------------");
    System.out.println("ENTER LEVEL :");
```

```
System.out.println("LEVEL 1: 0-29"+ '%' + " "+ '|'+"LEVEL 2: 30-39"+'%'+ " "+"|"+"LEVEL 3: 40-49"+'%'+" "+'|'+"LEVEL 4: 50-59"+'%'+" "+'|'+"LEVEL 5: 60-69"+'%'+" "+'|'+"LEVEL 6: 70-79"+'%'+" "+'|'+"LEVEL 7: 80-100"+'%');

studentGrade = inputs.nextLine();

System.out.println("--------------------------");

System.out.println("ENTER COURSE OF STUDY:");

System.out.println("MECHANICAL ENGINEERING: CREDITS 5"+"|"+"BEAUTY THERAPY: CREDITS 3"+"|"+"ELECTRICAL ENGINEERING: CREDITS 5"+"|"+"COMPUTER SCIENCE: CREDITS 4"+"|"+"JOURNALISM: CREDITS 3"+"|"+"HUMAN RESOURCES: CREDITS 4"+"|"+"COMPUTER STUDIES: CREDITS 4"+"|");

studentCourse = inputs.nextLine();

System.out.println("--------------------------");

System.out.println("ENTER DEPARTMENT");

System.out.println("ENGINEERING"+ "|"+"COMERCE"+ "|"+"MEDIA"+ "|"+"INFOMATION TECHNOLOGY"+ "|");

department = inputs.nextLine();

System.out.println("--------------------------");

System.out.println("ENTER DATE OF BIRTH ");

System.out.println("YYYY-MM-DD");

dateOfBirth = inputs.nextLine();

System.out.println("--------------------------");

System.out.println("ENTER EMAIL ADDRESS");

emailAddress = inputs.nextLine();

System.out.println("--------------------------");

System.out.println("ENTER PHONE NUMBER");

phoneNumber = inputs.nextLine();

System.out.println("--------------------------");

System.out.println("ENTER ADDRESS");

address = inputs.nextLine();

System.out.println("--------------------------");

System.out.println("ENTER ENROLLMENT DATE");

System.out.println("YYYY-MM-DD");

enrollmentDate = inputs.nextLine();

System.out.println("--------------------------");


try(FileWriter writer = new FileWriter(STUDENTINFO, true)){
```

```
        writer.write("Name: "+studentName +  " ||" + "Surname: "+studentSurname + " ||"+"Age: "+studentAge + " ||"
+"Level:" + studentGrade +  "||" +"Course:" +studentCourse +  "||" +"Date of Birth:" +dateOfBirth +  "||" +"Email Adress:"
+emailAddress +  "||" +"Phone Number:" +phoneNumber + "||"+"Adress:"+address+"||"+"Enrollment Date"+
enrollmentDate +"\n");

        System.out.println("***Information captured successfully into system***");

      } catch (IOException e){

        System.out.println("Error!!! Information capturing is unsuccessful: " + e.getMessage());

      }

   }

   static void readFromSystem(){

      try(BufferedReader reader = new BufferedReader(new FileReader(STUDENTINFO))){

        System.out.println("Student Infomation Caputed In System:");

        String line;

        while((line = reader.readLine()) != null){

           System.out.println(line);

        }

             }catch (IOException e){

             System.out.println("Error Retrieving Student Infomation:" + e.getMessage());

      }

   }


    static void system() {

   Scanner k = new Scanner(System.in);

   while(true){

      System.out.println("================================================");

      System.out.println("WELCOME TO STUDENT MANAGEMENT SYSTEM");

      System.out.println("================================================");

      System.out.println("1. Enter To Update Details if Current Student ");

      System.out.println("2. Enter To Student Registration if New Student");

      System.out.println("3. Enter To Log Out Of System");

      System.out.println("================================================");

      System.out.println("SYSTEM---ADMINISTRATOR--STATUS--LOGGED---IN");

      System.out.println("================================================");

      int choice;
```

```
choice = k.nextInt();


switch(choice){
   case 1
      Scanner H = new Scanner(System.in);
      while(true){
         System.out.println("-----------------------------------------------------------------");
         System.out.println("--Updating Student Details Into System--");
         System.out.println("-----------------------------------------------------------------");
         System.out.println("1. Enter to Details");
         System.out.println("2. Enter to Retrieve Student Details Captured In System");
         System.out.println("3. Enter to Return to Home Menu");
         int task;
         task = H.nextInt();
         switch(task){
            case 1:
               writeStudentInfo();
            break;
            case 2:
               readFromSystem();
            break;
            case 3:
               System.out.println("---------------------------------");
               System.out.println("Returning to Home Menu");
               System.out.println("---------------------------------");
               system();
            break;
            default:
               System.out.println("Invalid Option... Please try again");
            break;
         }
      }
   case 2:
```

```
            System.out.println("--Registering New Student Into System--");
            Scanner G = new Scanner(System.in);
            while(true){
               System.out.println("1. Enter to Register Student ");
               System.out.println("2. Enter to Retieve Student List of Registered Students");
               System.out.println("3. Enter to Return to Home Menu");
               int option;
               option = G.nextInt();

switch(option){
                    case 1:
                     writeStudentInfo();
                    break;
                    case 2:
                      readFromSystem();
                      break;
                    case 3:
                       System.out.println("-----------------------------------------------");
                       System.out.println("RETURNING TO HOME MENU");
                       System.out.println("-----------------------------------------------");
                       system();
                       break;
                    default:
                       System.out.println("Invalid Option. Please try again.");
                    break;

               }
            }
          case 3:
            System.out.println("Exting.....");
            System.out.println("-------------------------------------------------");
            System.out.println("SYSTEM---ADIMNISTAROR--STATUS--LOGGED---OUT");
            System.out.println("-------------------------------------------------");
```

```
            System.exit(0);
        }
    }


}
    public static void main(String[] args) {
        System.out.println("======================================================================");
        System.out.println("WELCOME TO STUDENT MANAGEMENT SYSTEM");
        System.out.println("======================================================================");
        UserLogin()
  }


}
```


```
SYSTEM---ADMINISTRATOR--STATUS--LOGGED---IN
1
--Updating Student Details Into System--
1. Enter to Details
2. Enter to Retrieve Student Details Captured In System
3. Enter to Return to Home Menu
1
ENTER NAME:
USE CAPITAL LETTERS
booob
ENTER SURNAME
USE CAPITAL LETTERS
sderf
ENTER AGE:
34
ENTER LEVEL :
LEVEL 1: 0-29% |LEVEL 2: 30-39% |LEVEL 3: 40-49% |LEVEL 4: 50-59% |LEVEL 5: 60-69% |LEVEL 6: 70-79% |LEVEL 7: 80-100%
2
ENTER COURSE OF STUDY:
```

```
LEVEL 1: 0-29% |LEVEL 2: 30-39% |LEVEL 3: 40-49% |LEVEL 4: 50-59% |LEVEL 5: 60-69% |LEVEL 6: 70-79% |LEVEL 7: 80-100%
2
---------------------------
ENTER COURSE OF STUDY:
MECHANICAL ENGINEERING: CREDITS 5|BEAUTY THERAPY: CREDITS 3|ELECTRICAL ENGINEERING: CREDITS 5|COMPUTER SCIENCE: CREDITS 4|JOURNALISM: CREDITS 3|HUMAN RESOURCES: CR
journalism
---------------------------
ENTER DEPARTMENT
ENGINEERING|COMERCE|MEDIA|INFOMATION TECHNOLOGY|
media
---------------------------
ENTER DATE OF BIRTH
YYYY-MM-DD
2022-04-12
---------------------------
ENTER EMAIL ADDRESS
real@gmail.com
---------------------------
ENTER PHONE NUMBER
0987466758
---------------------------
ENTER ADDRESS
12 Vukuthu Stress
---------------------------
ENTER ENROLLMENT DATE
YYYY-MM-DD
12 Vukuthu Stress
---------------------------
ENTER ENROLLMENT DATE
YYYY-MM-DD
2024-08-23
---------------------------
***Information captured successfully into system***
------------------------------------------------------------------
--Updating Student Details Into System--
------------------------------------------------------------------
1. Enter to Details
2. Enter to Retrieve Student Details Captured In System
3. Enter to Return to Home Menu
2
Student Infomation Caputed In System:
Name: LUCKY ||Surname: SITHOLE ||Age: 22 ||Level:5||Course:ELECTRICAL ENGINEERING||Date of Birth:2003-08-18||Email Adress:luckysithole03@gmail.com||Phone Number:0
Name: LEBO ||Surname: KIM ||Age: 24 ||Level:4||Course:BEAUTY THERAPY||Date of Birth:2001-06-23||Email Adress:lebo07@gmail.com||Phone Number:0824581245||Adress:45
Name: THATO ||Surname: BLUE ||Age: 20 ||Level:7||Course:HUMAN RESOURCE||Date of Birth:2005-07-12||Email Adress:thatoblue@gmail.com||Phone Number:0764537601||Adres
Name: LESEGO ||Surname: STEEZY ||Age: 20 ||Level:6||Course:COMPUTER SCIENCE||Date of Birth:2005-12-16||Email Adress:steezy707@gmail.com||Phone Number:06511745481|
Name: BAFANA ||Surname: JOBE ||Age: 23 ||Level:2||Course:MECHANICAL ENGINEERING||Date of Birth:2002-02-01||Email Adress:jobe204@gmail.com||Phone Number:0845245481
Name: PINKY ||Surname: WHITE ||Age: 22 ||Level:5||Course:BEAUTY THERAPY||Date of Birth:2003-05-11||Email Adress:pinkywhite@gmail.com||Phone Number:0713524537||Adr
Name: THANDO ||Surname: NKOSI ||Age: 21 ||Level:4||Course:JOURNALISM||Date of Birth:2004-02-14||Email Adress:thandonkosi5@gmail.com||Phone Number:0672510090||Adre
Name: KHANYA ||Surname: MAQAQA ||Age: 22 ||Level:1||Course:ELECTRICAL ENGINEERING||Date of Birth:2003-01-07||Email Adress:maqaqakhanya01@gmail.com||Phone Number:0
Name: NALEDI ||Surname: MOFOKENG ||Age: 20 ||Level:7||Course:BEAUTY THERAPY||Date of Birth:2005-01-15||Email Adress:naledimofokeng7@gmail.com||Phone Number:071453
Name: booob ||Surname: sderf ||Age: 34 ||Level:2||Course:journalism||Date of Birth:2022-04-12||Email Adress:real@gmail.com||Phone Number:0987466758||Adress:12 Vuk
------------------------------------------------------------------
--Updating Student Details Into System--
```

```
 Updating Student Details Into System
--------------------------------------------------------------------
1. Enter to Details
2. Enter to Retrieve Student Details Captured In System
3. Enter to Return to Home Menu
3
---------------------------------
Returning to Home Menu
---------------------------------
================================================
WELCOME TO STUDENT MANAGEMENT SYSTEM
================================================
1. Enter To Update Details if Current Student
2. Enter To Student Registration if New Student
3. Enter To Log Out Of System
================================================
SYSTEM---ADMINISTRATOR--STATUS--LOGGED---IN
================================================
2
--Registering New Student Into System--
1. Enter to Register Student
2. Enter to Retieve Student List of Registered Students
3. Enter to Return to Home Menu
1
--------------------------
ENTER NAME:
USE CAPITAL LETTERS
```

```
================================================
1. Enter To Update Details if Current Student
2. Enter To Student Registration if New Student
3. Enter To Log Out Of System
================================================
SYSTEM---ADMINISTRATOR--STATUS--LOGGED---IN
================================================
2
--Registering New Student Into System--
1. Enter to Register Student
2. Enter to Retieve Student List of Registered Students
3. Enter to Return to Home Menu
1
--------------------------
ENTER NAME:
USE CAPITAL LETTERS
chris
--------------------------
ENTER SURNAME
USE CAPITAL LETTERS
whyne
--------------------------
ENTER AGE:
31
--------------------------
ENTER LEVEL :
LEVEL 1: 0-29% |LEVEL 2: 30-39% |LEVEL 3: 40-49% |LEVEL 4: 50-59% |LEVEL 5: 60-69% |LEVEL 6: 70-79% |LEVEL 7: 80-100%
```

```
MECHANICAL ENGINEERING: CREDITS 5|BEAUTY THERAPY: CREDITS 5|ELECTRICAL ENGINEERING: CREDITS 5|COMPUTER SCIENCE: CREDITS 4|JOURNALISM: CREDITS 5|HUMAN RESOURCES: C
computer science
---------------------------
ENTER DEPARTMENT
ENGINEERING|COMERCE|MEDIA|INFOMATION TECHNOLOGY|
it
---------------------------
ENTER DATE OF BIRTH
YYYY-MM-DD
2004-01-23
---------------------------
ENTER EMAIL ADDRESS
jdh
---------------------------
ENTER PHONE NUMBER
295048
---------------------------
ENTER ADDRESS
gdfdrg
---------------------------
ENTER ENROLLMENT DATE
YYYY-MM-DD
2-3044
---------------------------
***Information captured successfully into system***
1. Enter to Register Student
2. Enter to Retieve Student List of Registered Students
3. Enter to Return to Home Menu
```

m,

```
ENTER ENROLLMENT DATE
YYYY-MM-DD
2-3044
---------------------------
***Information captured successfully into system***
1. Enter to Register Student
2. Enter to Retieve Student List of Registered Students
3. Enter to Return to Home Menu
3
---------------------------------------------
RETURNING TO HOME MENU
---------------------------------------------
=============================================
WELCOME TO STUDENT MANAGEMENT SYSTEM
=============================================
1. Enter To Update Details if Current Student
2. Enter To Student Registration if New Student
3. Enter To Log Out Of System
=============================================
SYSTEM---ADMINISTRATOR--STATUS--LOGGED---IN
=============================================
3
Exting.....
---------------------------------------------
SYSTEM---ADIMNISTAROR--STATUS--LOGGED---OUT
---------------------------------------------
BUILD SUCCESSFUL (total time: 10 minutes 25 seconds)
```

x

System Testing & Maintenance Report

Author: Fikile Masilo

Department Members: Teboho Sepanya

Week 1:

During Week 1, our team focused on selecting a suitable system topic for the project. We also assigned roles and responsibilities to each team member based on individual strengths and interests. In addition, we reviewed the project requirements to ensure a clear understanding of the expected outcomes. We also discussed our approach to testing the system and outlined a basic strategy for identifying and fixing any potential bugs that may arise during development.

Week 2:

We began by testing the programming code; however, nothing executed successfully during the initial run. At this stage, we have only started working on the programming aspect of the project. Moving forward, we still need to focus on developing and integrating the database component to ensure the system functions as intended.

Week 3/4:

During Weeks 3 and 4, there were no major developments on the project, as most of the team was focused on preparing for our DP3 assessments. Our group leader understood the academic pressure and allowed us time to study. However, we remained in communication throughout this period, occasionally sharing ideas and discussing our plans for continuing the project once our exams were completed.

Week 5:

We have tested the programming code, and while it runs successfully, there are still several areas that require improvement and adjustment. One of the key changes we identified is replacing the current “enter symbol” with a more user-friendly input that allows users to specify the year they are currently studying. Additionally, we need to implement a feature that

enables users to input their student information. For users who are not yet registered students, the system should offer a registration option.

We have also decided to incorporate a login feature for existing students, while new users will be directed to a registration page. Once the database is fully developed, it should integrate smoothly with the programming code to ensure that all stored data functions correctly within the system. Furthermore, we need to revise our .txt file to ensure it accurately saves and stores all the necessary information as intended.

Week 6:

This week, our primary focus was on finalizing all components of the project and resolving any outstanding issues. We dedicated time to thoroughly reviewing the system to ensure that every functionality operates as intended. One of the major milestones we achieved was the successful implementation of the login system, which is now fully functional and performing as expected.

In addition, we addressed and resolved issues related to our txt file, ensuring that it is properly formatted and integrated into the system. Our database has also been fully completed; we successfully normalized it to the appropriate forms and captured all the necessary data accurately. This has improved the efficiency and integrity of the data handling within our system

We conducted extensive testing across all modules of the application to identify and fix potential bugs. Based on the results, all features are working seamlessly, and the system is stable.

With all critical components implemented and tested, we are confident that the system is ready for deployment

PROGRAMMING REPORT

Author: Khanya Maqaqa

Department Members: Khanya Maqaqa

WEEK ONE.

In the first week, I conducted research on how to code a TXT file to create a TXT file. We did more research on how a TXT file works, how to manipulate a TXT file, and how to, in general, implement code around creating a TXT file. A simple student management system was created to give us a refence, It stores information as variables and prints them out, as it would allow you to take user input and just print out without storing in a TXT file.

WEEK TWO.

In week two, we came up with a prototype system where the system was able to write inside the TXT file and you could retrieve information from the TXT file and read.

Issues came with not being able, managing to store more than multiple student information, so we had to conduct more research on how to write multiple lines of data inside the TXT file. And with collaboration with system testing department, they made recommendations of how to make the interface more user friendly, as they felt the users would be confused with the interface.

Week THREE

Not much progress was done. I still trying to figure out how to write multiple students inside the TXT file, as the code that we came up with had issues of only writing one line inside the TXT file. And when you would add a new student information, it would remove the previous information you had written and replace it with the current one that you are putting in.

WEEK FOUR

I found solutions so the system can TXT file could write multiple, more than one line of students. We made adjustments and improvements on the interface, and it was recommend

that it would be advisable to implement a login into the system by the system testing and maintenance department.

WEEK FIVE TO WEEK SIX,

I was trying to implement the login, which was very much difficult. We ended up having two codes, where in the first code, the login was done in an if statement, where it would say, the user password and username inside an if statement, if it would allow the user to access the system.

And the second code, the second system we came up with, the login mechanism was done inside a TXT file, where if the information was found inside the TXT file, then it would allow the user to access it. In week six, a week, a final week, where we had to just change a lot of the interface, make it user friendly, focus more on the design of the system, to make it more visually appealing, as group members decided that it is important to at least make, to think about the user that would use the system to not be confused

DATABASE MANAGEMENT REPORT

Author: Dimakatso Yuweni

Written: 2025-06-18

Database Management Members: Lesego Mbambo, Lucky Sithole

Student Management System Database Report

The student management system has been designed to store and manage information related to students, courses, enrolments and registration, within an institution such as Gauteng City college. This database supports information such as tracking students, as well managing course offerings. It keeps or rather stores Student enrolments and registration information which can be accessed at any time by the student and the institution.

The purpose of this database is to provide information for students and courses information that helps with retrieving information, such as Updates and reports.

It helps with making sure that the student portal or rather system runs smoothly.

The problems that we encountered while creating Student the database for the system management system was the redundancy where we were repeating the same data in multiple tables. which was making the database inconsistent. We managed to fix the problem by normalizing the database design to lessen the data

redundancy. We also used separate tables for entities like Students, courses and enrolments etc.

We also encountered a problem such as incorrect or incomplete data entry like missing dates and. So to Solve the problem we used NULL and UNIQUE in our database.

Performance Report

Teboho Sepanya (System Testing & Maintaince)

Rea positively contributed to the group. I genuinely believed she could have been assigned to another department, either database or programming, as she possesses great skills. Because she lacked a laptop, she chose to work in the system testing and maintenance department. She performed her duties exceptionally in that department. In a team environment, she was open to sharing her thoughts and stepping beyond her duties to assist in various areas, especially the database department. The database department showcased a disorganized and chaotic database but was eager to improve and create clear and logical tables. As the leader of the group, I valued her attempts to help the team reach its objectives.

Fikile Masilo (System Testing & Mantaince)

Fikile was the most dedicated team member, as she made sure she never skipped a day of school while the team was still engaged in the project. She was consistently there and frequently contributed ideas that assisted the group in creating a sophisticated student management system. She rarely complained and consistently dedicated herself to whatever the group chose, even if she did not personally concur with them. One thing I would complain about Fikile sometimes is that she should communicate more outside of the school setting

Brandon Petersen(System Design & Maintain)

Brandon is the most optimistic member of the group. I had never been disheartened, even when I thought the group might not succeed. Throughout each phase of the student management system's development, Brandon consistently sought to stay informed about the progress. He approached his role with great seriousness and demonstrated a genuine enthusiasm for his work. He exceeded the expectations of what he was meant to do. Upon noticing that Tshiamo lacked a system design, Brandon had already provided one without being prompted. He is the most important member of the team because he has greatly supported me as a leader, particularly on a personal level, and truly motivates me to continue and guide our group

Dimakatso Yuweni (Database)

Dimakatso was always eager to take part in the group. When it came to assigning individuals to departments and distributing tasks, it should be noted that many avoided roles demanding extensive coding, yet she boldly volunteered and opted to engage with the database. Her personal life prevented her from attending regularly, but ensured that she participated even though she was away from. I wish she have more transparent about her challenges so that others could offer assistance with her tasks. It makes sense that she didn't possess a laptop at home. I believe she may not have been the most proficient in the database, but her enthusiasm is the most admirable trait she possesses.

Lesego Mbambo (Database)

Lesego isn't the strongest in communication, as I think he has room for improvement in that area. He certainly took part in developing the database. I wish he had been more open, particularly when he misses school, as his regular communication would have reassured everyone. I discovered that his access to a functioning laptop was restricted, yet he refused to let the group down. His optimism impacted me since he didn’t let negativity influence him in the group interactions. I realize he preferred not to complain, but I truly wish he had because I would have ensured that changes were implemented to deliver an excellent database. The database department overall did not produce an end product that was satisfactory.

Lucky Sithole (Database)

Initially, Lucky worked in the programming department. His responsibility was to participate in developing the system. I recognize that programming is difficult, but he truly didn't put in any effort. I would take my laptop to school, but he was never involved in the programming aspect. He would inevitably find himself doing something else. He constantly gives me the impression that he would assist, but he never does. I observed that he would mention going home to work on the system, particularly when attempting to comprehend how to code for the txt file and the login process. He did not succeed in making progress. I wish he had informed me of his

limitations so I could have tailored my approach to his strengths. He contributed to the database. He significantly assisted that department in producing an end product, as the database department struggled for 5 weeks to even show a prototype database. He entered and genuinely supported the database department. I truly wish he had been more truthful with me so that he could have received a position he might have excelled in

Sarah Semenya (Failed to Participated)

Sarah truly let everyone down. She was rarely at school. She participated in a single group meeting. Her private life significantly influenced her involvement. Her responsibility was to create the system by developing data flow diagrams and entity relationship diagrams. She remained silent the whole time, aware and consistently stated she would complete her tasks. As the group leader, I was ready to help her, but she never communicated her issues to me. I believe her problem arose from collaborating with someone she didn't know well, . I believe if I had matched her with Dimakatso, she might have been more motivated to complete her tasks.