**Project Report**

**On**

**UNIVERSITY COURSE MANAGEMENT**

**Session 2018-19**

**BTECH CSE**

**Submitted By**

**YATIN AGGARWAL**

**1803210184**

**Under the guidance of**

**Mr. TARUN KUMAR SHARMA**

**ABES ENGINEERING COLLEGE, GHAZIABAD**

**AFFILIATED TO**

**DR. A.P.J. ABDUL KALAMTECHNICAL UNIVERSITY, U.P., LUCKNOW (Formerly UPTU)**

**STUDENT’S DECLARATION**

I / We hereby declare that the work being presented in this report entitled “UNIVERSITY COURSE MANAGEMENT” is an authentic record of my own work carried out under the supervision of Mr. “TARUN KUMAR SHARMA”

The matter embodied in this report has not been submitted by me for the award of any other degree.

**Dated: Signature of students(s)**

**(Name(s).......................)**

**Department:**

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

**Signature of Supervisor**

**TABLE OF CONTENTS** Page

DECLARATION ................................................................................................... 2

CHAPTER 1 4

1.1.Problem Introduction

1.1.1.Motivation..................................................................... 4

1.1.2. Project Objective............................................................ 4

1.1.3Scope of the Project........................................................ 4

1.2Introduction to Technologies

1.2.1.Language.................................................................. 4

1.2.2.Library/ In-built function / package / modules...................... 5

1.2.3.Hardware Requirement .................................................... 5

1.2.4.Software Requirement................................................... 6

1.2.5.IDE............................................................................ 6

CHAPTER 2

2.1Algorithm/ Approach.............................................................. 7

2.2 Work Flow Diagram................................................................ 7

2.3 Explain your Code.................................................................. 8

CHAPTER 3

3.1Snapshots / Output of Interfaces................................................ 17

CHAPTER 4 (CONCLUSIONS) ..................................................................... 20

REFERENCES... ........................................................................... 20

**CHAPTER 1**

**INTRODUCTION**

A University Course Management System is a Management System which helps the University to easily keep its records and retrieve data in an organised and easy way with any filter as per the requirement.

A management system is a set of [policies](https://en.wikipedia.org/wiki/Policy), [processes](https://en.wikipedia.org/wiki/Business_process) and [procedures](https://en.wikipedia.org/wiki/Procedure_(term)) used by an [organization](https://en.wikipedia.org/wiki/Organization) to ensure that it can fulfil the tasks required to achieve its objectives.

1.1. A university database contains information about professors (identified by social security number, or SSN) and courses (identified by course\_id).

Professors teach courses; each of the following situations concerns the Teaches relationship set. Go through the following conditions

1. Professors can teach the same course in several semesters, and each offering must be recorded.

2. Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded (assume this condition applies in all subsequent questions)

3. Every professor must teach some course.

4. Every professor teaches exactly one course.

5. Every professor teaches exactly one course and every course must be taught by some professor.

6. Now suppose that certain courses can be taught by the team of professors jointly, but it is possible that no one professor in a team can teach the course

1.1.1. The motivation for doing this project was primarily an interest in undertaking a challenging project in an interesting area of research. The opportunity to learn about a new area of computing not covered in lectures was appealing.

1.1.2.To make a University Database

1.1.3. All Universities

1.2.Introduction to Technologies

1.2.1. Java

Java is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_language) [programming language](https://en.wikipedia.org/wiki/Programming_language) that is [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) (although not a pure OO language, as it contains primitive types), and designed to have as few implementation [dependencies](https://en.wikipedia.org/wiki/Dependency_(computer_science)) as possible. It is intended to let [application developers](https://en.wikipedia.org/wiki/Application_developer) write once, run anywhere (WORA), meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of the underlying [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). The [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) of Java is similar to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them. As of 2018, Java was one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) according to [GitHub](https://en.wikipedia.org/wiki/GitHub), particularly for [client-server](https://en.wikipedia.org/wiki/Client%E2%80%93server) [web applications](https://en.wikipedia.org/wiki/Web_applications), with a reported 9 million developers.

Java was originally developed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) ([which has since been acquired by Oracle](https://en.wikipedia.org/wiki/Sun_acquisition_by_Oracle)) and released in 1995 as a core component of Sun Microsystems' [Java platform](https://en.wikipedia.org/wiki/Java_(software_platform)). The original and [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) Java [compilers](https://en.wikipedia.org/wiki/Compiler), virtual machines, and [class libraries](https://en.wikipedia.org/wiki/Library_(computing)) were originally released by Sun under [proprietary licenses](https://en.wikipedia.org/wiki/Proprietary_license). As of May 2007, in compliance with the specifications of the [Java Community Process](https://en.wikipedia.org/wiki/Java_Community_Process), Sun had [relicensed](https://en.wikipedia.org/wiki/Software_relicensing) most of its Java technologies under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License). Meanwhile, others have developed alternative implementations of these Sun technologies, such as the [GNU Compiler for Java](https://en.wikipedia.org/wiki/GNU_Compiler_for_Java) (bytecode compiler), [GNU Classpath](https://en.wikipedia.org/wiki/GNU_Classpath) (standard libraries), and [IcedTea](https://en.wikipedia.org/wiki/IcedTea)-Web (browser plugin for applets).

1.2.2Library/ In-built function / package / modules used in your project

*Java*.*util Package*. It contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string tokenizer, a random-number generator, and a bit array).

1.2.3.Hardware Requirement

IBM-compatible 486 system.

Hard Drive and Minimum of 8 MB memory.

Mouse, keyboard and sound card, if required.

1.2.4.Software Requirement

Minimum Windows 95 software.

1.2.5.IDE

Eclipse is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming), and in 2014 was the most widely used Java IDE in one website's poll. It contains a base [workspace](https://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](https://en.wikipedia.org/wiki/Plug-in_(computing)) system for customizing the environment. Eclipse is written mostly in [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) and its primary use is for developing Java applications, but it may also be used to develop applications in other [programming languages](https://en.wikipedia.org/wiki/Programming_language) via plug-ins, including [Ada](https://en.wikipedia.org/wiki/Ada_(programming_language)), [ABAP](https://en.wikipedia.org/wiki/ABAP), [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [Clojure](https://en.wikipedia.org/wiki/Clojure), [COBOL](https://en.wikipedia.org/wiki/COBOL), [D](https://en.wikipedia.org/wiki/D_(programming_language)), [Erlang](https://en.wikipedia.org/wiki/Erlang_(programming_language)), [Fortran](https://en.wikipedia.org/wiki/Fortran), [Groovy](https://en.wikipedia.org/wiki/Groovy_(programming_language)), [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), [Julia](https://en.wikipedia.org/wiki/Julia_(programming_language)), [Lasso](https://en.wikipedia.org/wiki/Lasso_(programming_language)), [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)), [NATURAL](https://en.wikipedia.org/wiki/Software_AG), [Perl](https://en.wikipedia.org/wiki/Perl), [PHP](https://en.wikipedia.org/wiki/PHP), [Prolog](https://en.wikipedia.org/wiki/Prolog), [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [R](https://en.wikipedia.org/wiki/R_(programming_language)), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)) (including [Ruby on Rails](https://en.wikipedia.org/wiki/Ruby_on_Rails) framework), [Rust](https://en.wikipedia.org/wiki/Rust_(programming_language)), [Scala](https://en.wikipedia.org/wiki/Scala_(programming_language)), and [Scheme](https://en.wikipedia.org/wiki/Scheme_(programming_language)). It can also be used to develop documents with [LaTeX](https://en.wikipedia.org/wiki/LaTeX) (via a TeXlipse plug- in) and packages for the software [Mathematica](https://en.wikipedia.org/wiki/Mathematica). Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial [codebase](https://en.wikipedia.org/wiki/Codebase) originated from [IBM VisualAge](https://en.wikipedia.org/wiki/IBM_VisualAge). The Eclipse [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since the introduction of the [OSGi](https://en.wikipedia.org/wiki/OSGi) implementation ([Equinox](https://en.wikipedia.org/wiki/Equinox_(OSGi))) in version 3 of Eclipse, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundles Eclipse [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) (SDK) is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software), released under the terms of the [Eclipse Public License](https://en.wikipedia.org/wiki/Eclipse_Public_License), although it is incompatible with the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License). It was one of the first IDEs to run under [GNU Classpath](https://en.wikipedia.org/wiki/GNU_Classpath) and it runs without problems under [IcedTea](https://en.wikipedia.org/wiki/IcedTea).

**CHAPTER 2**

**SYSTEM DESIGN**

**2.1 Algorithm/ Approach**

It is a menu driven program having 3 entities Professor, Student & Subject.

After adding Professors, Students and Subjects the allotted Professors or a team of Professors is displayed as a result

In Professor we have to add the number of Professors

In Students we have to add the number of Students

In Subjects we have to add the number of Subjects

**2.2 Work Flow Diagram**

It is a menu driven program having 3 entities Professor, Student & Subject In Professor we have to add the number of Professors

In Students we have to add the number of Students

In Subjects we have to add the number of Subjects

In Result different Professors are allotted to Courses based on the conditions specified in the problem and are displayed as the final output

**Sample Work flow**

**2.3.Explain your Code**

**import** java.util.\*;

**public** **class** UniversityCourseManagement {

**public** **static** **void** main(String args[]) {

Menu.*menuFunction*();

}

}

**class** Menu {

**static** **void** mainMenu() {

System.***out***.println("\nUniversity Course Management");

System.***out***.println("1. Professor");

System.***out***.println("2. Student");

System.***out***.println("3. Subject");

System.***out***.println("4. Result");

System.***out***.println("5. Exit");

System.***out***.print("Enter your choice : ");

}

**static** **void** menuFunction() {

**int** x=0; //choice of menu

Scanner sc = **new** Scanner(System.***in***);

*mainMenu*();

x = sc.nextInt();

**switch**(x) {

**case** 1:

Professor.*professorMenu*();

**break**;

**case** 2:

Student.*studentMenu*();

**break**;

**case** 3:

Subject.*subjectMenu*();

**break**;

**case** 4:

Result.*result*();

**break**;

**case** 5:

System.*exit*(0);

}

}

}

**class** Professor {

**public** **static** String *profId*[] = **new** String[100];

**public** **static** String *profName*[] = **new** String[100];

**public** **static** **int** *profOfStudent*[] = **new** **int**[100];

**public** **static** **int** *n*=0;

**static** **void** professorMenuData() {

System.***out***.println("\nUniversity Course Management");

System.***out***.println("Professor");

System.***out***.println("1. Add Professor");

System.***out***.println("2. Delete Professor");

System.***out***.println("3. Display Professors");

System.***out***.println("4. Edit Professor");

System.***out***.println("5. Back");

System.***out***.println("6. Exit");

System.***out***.print("Enter your choice : ");

}

**static** **void** addProfessor() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter Professor's ID : ");

*profId*[*n*] = sc.nextLine();

System.***out***.print("Enter Professor's name : ");

*profName*[*n*] = sc.nextLine();

*n*++;

}

**static** **void** deleteProfessor() {

String temp;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter Professor's ID : ");

temp = sc.nextLine();

**for**(**int** i=0;i<*n*;i++) {

**if**(*profId*[i].contentEquals(temp)) {

**for**(;i<*n*-1;i++) {

*profId*[i]=*profId*[i+1];

*profName*[i]=*profName*[i+1];

}

*n*--;

**break**;

}

}

}

**static** **void** displayProfessor() {

System.***out***.println("\n Professor Data");

System.***out***.println("ID Name");

**for**(**int** i=0;i<*n*;i++) {

System.***out***.println(*profId*[i] + " " + *profName*[i]);

}

System.***out***.println();

}

**static** **void** editProfessor() {

String temp;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter Professor's ID : ");

temp = sc.nextLine();

**for**(**int** i=0;i<*n*;i++) {

**if**(*profId*[i].contentEquals(temp)) {

System.***out***.print("\nEnter Professor's Correct Name : ");

temp = sc.nextLine();

*profName*[i]=temp;

**break**;

}

}

}

**static** **void** professorMenu()

{

**int** x=0; //choice of Professor menu

Scanner sc = **new** Scanner(System.***in***);

*professorMenuData*();

x = sc.nextInt();

**switch**(x) {

**case** 1:

*addProfessor*();

Professor.*professorMenu*();

**break**;

**case** 2:

*deleteProfessor*();

Professor.*professorMenu*();

**break**;

**case** 3:

*displayProfessor*();

Professor.*professorMenu*();

**break**;

**case** 4:

*editProfessor*();

Professor.*professorMenu*();

**break**;

**case** 5:

Menu.*menuFunction*();

**case** 6:

System.*exit*(0);

}

}

}

**class** Student {

**public** **static** String *studId*[] = **new** String[100];

**public** **static** String *studName*[] = **new** String[100];

**public** **static** **int** *n*=0;

**static** **void** studentMenuData() {

System.***out***.println("\nUniversity Course Management");

System.***out***.println("Student");

System.***out***.println("1. Add Student");

System.***out***.println("2. Delete Student");

System.***out***.println("3. Display Student");

System.***out***.println("4. Edit Student");

System.***out***.println("5. Back");

System.***out***.println("6. Exit");

System.***out***.print("Enter your choice : ");

}

**static** **void** addStudent() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter Student's ID : ");

*studId*[*n*] = sc.nextLine();

System.***out***.print("Enter Student's name : ");

*studName*[*n*] = sc.nextLine();

*n*++;

}

**static** **void** deleteStudent() {

String temp;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter Student's ID : ");

temp = sc.nextLine();

**for**(**int** i=0;i<*n*;i++) {

**if**(*studId*[i].contentEquals(temp)) {

**for**(;i<*n*-1;i++) {

*studId*[i]=*studId*[i+1];

*studName*[i]=*studName*[i+1];

}

*n*--;

**break**;

}

}

}

**static** **void** displayStudent() {

System.***out***.println("\n Student Data");

System.***out***.println("ID Name");

**for**(**int** i=0;i<*n*;i++) {

System.***out***.println(*studId*[i] + " " + *studName*[i]);

}

System.***out***.println();

}

**static** **void** editStudent() {

String temp;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter Student's ID : ");

temp = sc.nextLine();

**for**(**int** i=0;i<*n*;i++) {

**if**(*studId*[i].contentEquals(temp)) {

System.***out***.print("\nEnter Student's Correct Name : ");

temp = sc.nextLine();

*studName*[i]=temp;

**break**;

}

}

}

**static** **void** studentMenu()

{

**int** x=0; //choice of Student menu

Scanner sc = **new** Scanner(System.***in***);

*studentMenuData*();

x = sc.nextInt();

**switch**(x) {

**case** 1:

*addStudent*();

Student.*studentMenu*();

**break**;

**case** 2:

*deleteStudent*();

Student.*studentMenu*();

**break**;

**case** 3:

*displayStudent*();

Student.*studentMenu*();

**break**;

**case** 4:

*editStudent*();

Student.*studentMenu*();

**break**;

**case** 5:

Menu.*menuFunction*();

**case** 6:

System.*exit*(0);

}

}

}

**class** Subject {

**public** **static** String *subId*[] = **new** String[100];

**public** **static** String *subName*[] = **new** String[100];

**public** **static** **int** *n*=0;

**static** **void** subjectMenuData() {

System.***out***.println("\nUniversity Course Management");

System.***out***.println("Subject");

System.***out***.println("1. Add Subject");

System.***out***.println("2. Delete Subject");

System.***out***.println("3. Display Subjects");

System.***out***.println("4. Edit Subject");

System.***out***.println("5. Back");

System.***out***.println("6. Exit");

System.***out***.print("Enter your choice : ");

}

**static** **void** addSubject() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter Subject ID : ");

*subId*[*n*] = sc.nextLine();

System.***out***.print("Enter Subject name : ");

*subName*[*n*] = sc.nextLine();

*n*++;

}

**static** **void** deleteSubject() {

String temp;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter Subject ID : ");

temp = sc.nextLine();

**for**(**int** i=0;i<*n*;i++) {

**if**(*subId*[i].contentEquals(temp)) {

**for**(;i<*n*-1;i++) {

*subId*[i]=*subId*[i+1];

*subName*[i]=*subName*[i+1];

}

*n*--;

**break**;

}

}

}

**static** **void** displaySubject() {

System.***out***.println("\n Subject Data");

System.***out***.println("ID Name");

**for**(**int** i=0;i<*n*;i++) {

System.***out***.println(*subId*[i] + " " + *subName*[i]);

}

System.***out***.println();

}

**static** **void** editSubject() {

String temp;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("\nEnter Subject ID : ");

temp = sc.nextLine();

**for**(**int** i=0;i<*n*;i++) {

**if**(*subId*[i].contentEquals(temp)) {

System.***out***.print("\nEnter Correct Subject Name : ");

temp = sc.nextLine();

*subName*[i]=temp;

**break**;

}

}

}

**static** **void** subjectMenu()

{

**int** x=0; //choice of Subject menu

Scanner sc = **new** Scanner(System.***in***);

*subjectMenuData*();

x = sc.nextInt();

**switch**(x) {

**case** 1:

*addSubject*();

Subject.*subjectMenu*();

**break**;

**case** 2:

*deleteSubject*();

Subject.*subjectMenu*();

**break**;

**case** 3:

*displaySubject*();

Subject.*subjectMenu*();

**break**;

**case** 4:

*editSubject*();

Subject.*subjectMenu*();

**break**;

**case** 5:

Menu.*menuFunction*();

**case** 6:

System.*exit*(0);

}

}

}

**class** Result{

**static** **int** calculate()

{

**if**(Subject.*n*==0 || Student.*n*==0 || Professor.*n*==0 || Professor.*n* < Subject.*n*\*Student.*n*)

**return** -1;

**else**

**return** Professor.*n* / (Subject.*n*\*Student.*n*);

}

**static** **void** result()

{

**int** n = *calculate*();

**if**(n<0) {

System.***out***.println("Not Possible");

}

**else** {

**int** i,j,k;

**int** m = Professor.*n* % (Subject.*n*\*Student.*n*);

System.***out***.println("Professor Subject Student");

**for**(i=0,k=0;i<Subject.*n*\*Student.*n*;i++) {

**if**(i<m) {

**for**(j=0;j<n+1;j++,k++) {

**if**(j != n)

System.***out***.print(Professor.*profId*[k] + " ");

**else**

System.***out***.print(Professor.*profId*[k] + " ");

}

}

**else** {

**for**(j=0;j<n;j++,k++) {

**if**(j != n-1)

System.***out***.print(Professor.*profId*[k] + " ");

**else**

System.***out***.print(Professor.*profId*[k] + " ");

}

}

System.***out***.println(Subject.*subId*[i%Student.*n*] + " " + Student.*studId*[i%Subject.*n*]);

}

System.***out***.println();

}

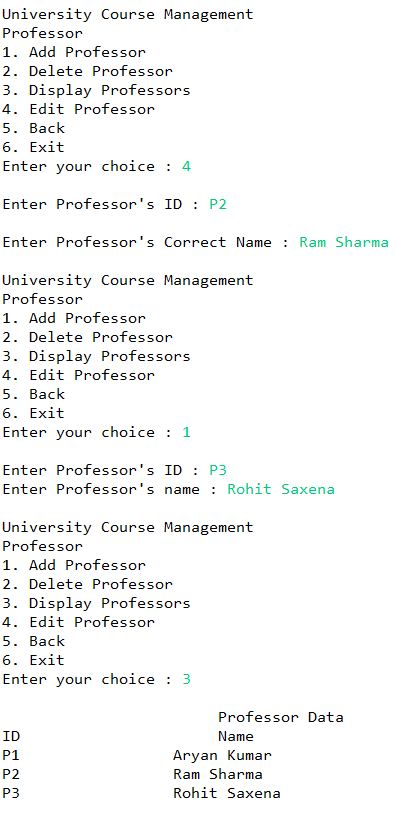
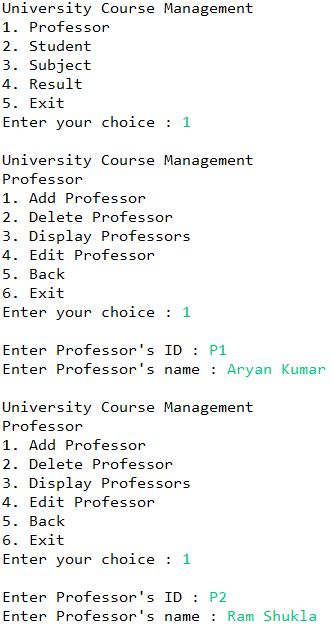
}

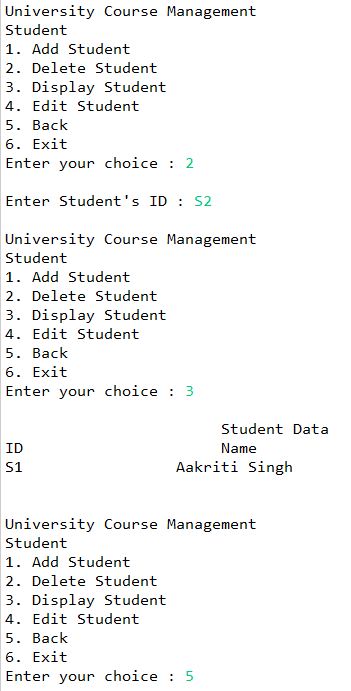
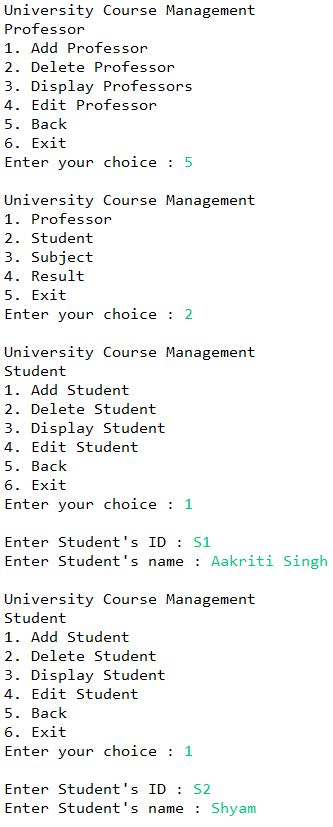
}

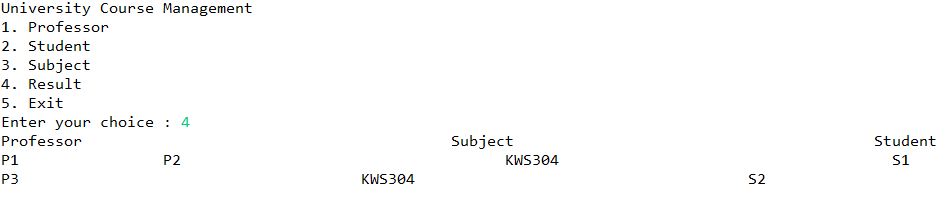
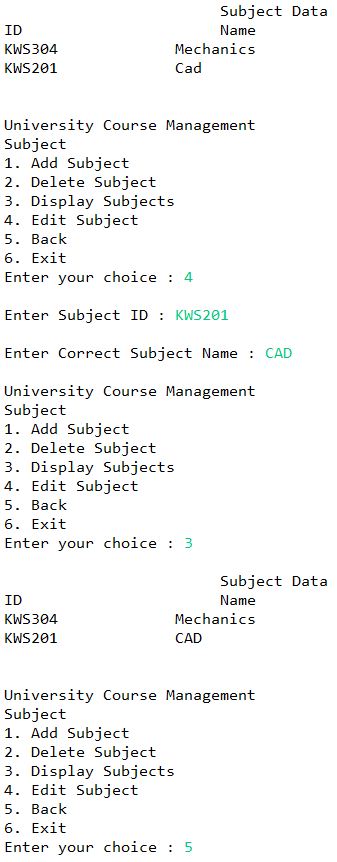
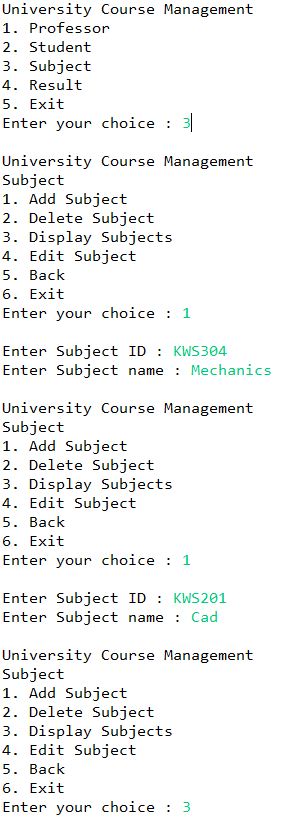
**CHAPTER 3**

**RESULTS**

**3. Snapshots / Output of Interfaces**

****

****

****

**CHAPTER 4**

**CONCLUSION**

I created a University Course Management System that a University can use for allocating the professors for the courses in different semesters according to the number of courses.

**References**

1. https://www.geeksforgeeks.org/java/

2. https://www.w3schools.com/java/

3. https://www.tutorialspoint.com/java/util/