Contents

[Introduction 2](#_Toc535057211)

[User of the System 2](#_Toc535057212)

[Functional Requirement 3](#_Toc535057213)

[Admin 3](#_Toc535057214)

[Student 3](#_Toc535057215)

[Tools to be used 3](#_Toc535057216)

[Entry class 3](#_Toc535057217)

[EntryForm Class 4](#_Toc535057218)

[Admin Login 4](#_Toc535057219)

[Admin Class 5](#_Toc535057220)

[Addmission Class 5](#_Toc535057221)

[Student\_Info class 6](#_Toc535057222)

[Information 6](#_Toc535057223)

[Search 7](#_Toc535057224)

[Setting 7](#_Toc535057225)

[PasswordChange 7](#_Toc535057226)

[Class Driver 8](#_Toc535057227)

[Constructor Summary 8](#_Toc535057228)

[Constructor and Description 8](#_Toc535057229)

[Method Summary 8](#_Toc535057230)

[Class CourseResult 8](#_Toc535057231)

[Constructor Summary 9](#_Toc535057232)

[Constructor and Description 9](#_Toc535057233)

[Method Summary 9](#_Toc535057234)

[**Modifier and Type** 9](#_Toc535057235)

[**Method and Description** 9](#_Toc535057236)

[java.lang.String 9](#_Toc535057237)

[Getter Method for Course ID 9](#_Toc535057238)

[java.lang.String 9](#_Toc535057239)

[Class UetGradeBook 10](#_Toc535057240)

[Constructor Summary 10](#_Toc535057241)

[Constructor and Description 10](#_Toc535057242)

[Method Summary 10](#_Toc535057243)

[**Modifier and Type** 10](#_Toc535057244)

[**Method and Description** 10](#_Toc535057245)

[double 10](#_Toc535057246)

[This Method returns the CGPA of the Student 10](#_Toc535057247)

[java.util.ArrayList<**CourseResult**> 10](#_Toc535057248)

[Getter Method for the list of Courses 10](#_Toc535057249)

[java.lang.String 10](#_Toc535057250)

[Getter method for Degree 10](#_Toc535057251)

[java.lang.String 10](#_Toc535057252)

[java.lang.String 11](#_Toc535057253)

[Getter Method for the Student Registration Number 11](#_Toc535057254)

[This Method returns the Total Credit Hours in a Semester Total is based on the number of registered Subjects 11](#_Toc535057255)

[StudentDetails 12](#_Toc535057256)

[StudentDetails 12](#_Toc535057257)

[StudentCourseDetails 13](#_Toc535057258)

# Introduction

Student management deals with all kinds of student details like adding new student, academic related details, course details, result details. It tracks all details of a student which can be used for all reporting purpose, personal information, result information, progresses in course etc.

My design can facilitate us to explore the current activities of a student. The student management system is an automated version of manual Student Management System. It can handle maximum details of a student. The details include subject details, personal details, academic details etc.

In case of manual system they need a lot of time and manpower etc. Here almost all work is computerized. So the accuracy is maintained. Maintaining backup is very easy. It can do with in a few minutes. My system has two types of accessing modes, administration and user. Student management system is managed by administration. It is the job of the administration to insert update and monitor the whole process. When a user log in to the system, he/she would only view his/her details. He/she can’t perform any changes.

# User of the System

1. Admin
2. Student

# Functional Requirement

# Admin

1. Can add student
2. Can view student information
3. Can add course
4. Can add result and semester
5. Search Student
6. Can change his/her own password

# Student

1. Can view personal information
2. Can view Completed course details
3. Can view CGPA

# Tools to be used

1. Use any ide to develop this project. It may be eclipse/ net beans etc.
2. File Handling for database

# Entry class

**Constructor Details:**

Entry()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

This is the first entry form. Here I have use ImageIcon to get the resource of the image and add the image into the Label. I also use a label for setting the heading information, welcoming title and quote. I have used a button for log in to the next page. Here I have use Handler class to implements actionlistener. Adding actionlistener and actionperformed in the button I got the access to login the next page.

# EntryForm Class

**Constructor**

EntryForm()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

This is the entry form for both administration and user. Here I have used two Radio button for both administration and user login process. I have used two button, one for individual login age and other for go to the previous page. . Here I have use Handler class to implements actionlistener. Adding actionlistener and actionperformed in the button and radio button, I got the access to go to the individual login page. E.g., If admin login radio button is selected and then submit button is pressed, then open a admin login page.

# Admin Login

**Constructor:**

AdminLogin()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

button-add action listener

This is the admin login page. Here I have used two jlables and two textfields. I have used three buttons named login, clear and back. In all three button I have added action listener. Here admin id is fixed and I have saved a password in a text file called “adminpassword.txt” for admin login. After setting jtextfield to the id and jpassword filed the password, then password will be check using bufferedreader. If the password match then there will be visible a pop up message and after pressing ok in the pop up message, we will be able to go to the next page. Clear button clear the textfield and password field. Back button is for go to the previous page. Here I have also add action listener in the textfield. If the user didn’t set any text to the text field and pressed login button there will visible a pop up message for didn’t entering anything in to the textfield.

# Admin Class

**Constructor:**

Admin()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

button-add action listener

This is the admin panel form. Here I have used two radio buttons and two buttons. I have added action listener to both buttons. For adding new student, users have to select admission radio button and press submit button, there will be visible admission form. For seeing details info of all the student, search for a particular student, result details and setting operation, users have to select student radio button and press submit button. Back button is for previous page.

# Addmission Class

**Constructor:**

Addmission()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

button-add action listener

This is the admission form. Here admin can add new student. From getting value from name, date of birth, phone number, address, father’s name, mother’s name, email, nationality I have used text field. For gender selection I have used radio buttons named male, female, others. For choosing department and blood group I have used combo box. In all three buttons, I have added action listener. If add student is clicked all the information will be added into a text file. Clear button will clear all the text in the text field. Back button is for previous page. Here I have added action listener in the entire text field. If any text field is blank and user press enter there will be show a pop up message. If any info is not set in the text field or any item is not selected, add student button can’t be added any info. For adding info in a text file I have used filewriter.

# Student\_Info class

**Constructor:**

Student\_Info()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

button-add action listener

This is student details information form. Here I have used three radio button and add action listener to the entire radio button using handler class. If information button is clicked then there will be visible a form where all students info will be shown. Search button is for searching particular student. Setting button is for changing password of the admin. Back button is for going to the previous page.

# Information

**Constructor:**

Information()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

button-add action listener

This form is for showing all the personal information of a student. For showing info I have generated a default table model. Using file reader and buffered reader I have read the text where all information of the students was added and inset the info into the table row. Then I added the row with the default table model. Previously I have inserted the columns name into array and declared the size of the array. The result button is for inserting result of a particular student. And back button is for previous page. I have added mouse listener into the table. Whenever I will select any row, the information will show into another form.

# Search

**Constructor:**

Search()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

button-add action listener

Here I have added action listener to search button. Here first of all I save the word of textfield and Then match the word with existing info file using buffered reader. If the word matches then all info of that name will show in the text area.

# Setting

# PasswordChange

**Constructor:**

Information()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

button-add action listener

For admin setting, admin can change his/her password. For changing password first of all, I read the current password using buffered reader and then match type and retype both password fields. If they matches then the existing “adminpassword.txt” file will override the new password.

# Class Driver

java.lang.Object

driver.Driver

public class **Driver**

extends java.lang.Object

## Constructor Summary

# Constructor and Description

Public Driver ()

## Method Summary

Main:

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

# Class CourseResult

java.lang.Object

driver.CourseResult

public class **CourseResult**

extends java.lang.Object

extends java.lang.Object

## Constructor Summary

# Constructor and Description

[CourseResult](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#CourseResult--)()

Default Constructor

[CourseResult](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#CourseResult-driver.CourseResult-)([CourseResult](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html" \o "class in driver) CourseRsult)

copy Constructor

[CourseResult](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#CourseResult-java.lang.String-java.lang.String-int-int-int-)(java.lang.String id, java.lang.String title, int CH, int sem, int marks)

constructor with parameters

## Method Summary

|  |  |
| --- | --- |
| **Modifier and Type** | **Method and Description** |
| java.lang.String | [**getCourseID**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#getCourseID--)()  Getter Method for Course ID |
| java.lang.String | [**getCourseTitle**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#getCourseTitle--)()  Getter Method for Course Title |
| Int | [**getCreditHours**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#getCreditHours--)()  getter method for Credit Hours |
| java.lang.String | [getGrade](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#getGrade--)()  function to get the grade of Subject |
| double | [**getGradePoints**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#getGradePoints--)()  member functions get the grade points |
| Int | [getMarks](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#getMarks--)()  getter method for Mark |
| Int | [getSemester](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#getSemester--)()  getter method for Semester |
| boolean | [setCourseID](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#setCourseID-java.lang.String-)(java.lang.String id)  Setter Method for Course ID |
| boolean | [setCreditHours](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#setCreditHours-int-)(int cHours)  Setter Method for Credit Hours |
| boolean | [**setMarks**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#setMarks-int-)(int Marks)  setter method for Marks |
| boolean | [**setSemester**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#setSemester-int-)(int Sem)  Setter Method for Semester |
| java.lang.String | [**toString**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html#toString--)()  overriding toString Method |

# Class UetGradeBook

java.lang.Object

driver.UetGradeBook

public class **UetGradeBook**

extends java.lang.Object

## Constructor Summary

# Constructor and Description

[**UetGradeBook**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#UetGradeBook--)()

Default Constructor

## Method Summary

|  |  |
| --- | --- |
| **Modifier and Type** | **Method and Description** |
| double | [**getCGPA**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getCGPA--)()  This Method returns the CGPA of the Student |
| [CourseResult](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html) | [**getCoursebyID**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getCoursebyID-java.lang.String-)(java.lang.String crsID)  This Method returns the reference Object with matching Course ID to update from List |
| java.util.ArrayList<**[CourseResult](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html" \o "class in driver)**> | [**getCourses**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getCourses--)()  Getter Method for the list of Courses |
| java.lang.String | [**getDegree**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getDegree--)()  Getter method for Degree |
| java.lang.String | [**getDescipline**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getDescipline--)()  This Function will return the name of the discipline on the basis of registration number |
| java.lang.String | [**getRegNumber**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getRegNumber--)()  Getter Method for the Student Registration Number |
| Int | [**getSemesterCreditHours**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getSemesterCreditHours-int-)(int semester)  This Method returns the Total Credit Hours in a Semester Total is based on the number of registered Subjects |
| double | [**getSemesterGPA**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getSemesterGPA-int-)(int semester)  This Function calculates and returns the GPA of the required semester number |
| Int | [**getSemesters**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getSemesters--)()  This Function Calculates the number of Semesters in which a student has registered s |
| Int | [**getSession**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getSession--)()  This Function will return the Session year on the basis of registration number |
| java.lang.String | [**getStudentName**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getStudentName--)()  Getter Method to for Name of the Student |
| Int | [**getTotalCreditHours**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#getTotalCreditHours--)()  This Function will return the total credit hours of all semesters Total is based on the number of registered Subjects |
| boolean | [**removeCoursebyID**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#removeCoursebyID-java.lang.String-)(java.lang.String crsID)  This Method removes a Course from List of Courses |
| boolean | [**setCourse**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#setCourse-driver.CourseResult-)(**[CourseResult](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html" \o "class in driver)** course)  Setter Method to Set the Course object into the ArrayList of Courses |
| Void | [**setCourses**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#setCourses-java.util.ArrayList-)(java.util.ArrayList<**[CourseResult](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/CourseResult.html" \o "class in driver)**> Courses)  setter for courses list |
| boolean | [**setDegree**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#setDegree-java.lang.String-)(java.lang.String department)  Setter Method for Degree Name Degree Name Can be CSE, BBA or EEE it is validated inside |
| boolean | [**setRegNumber**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#setRegNumber-java.lang.String-)(java.lang.String RegNo)  Setter Method for Student Registration Number it validates the Registration number |
| boolean | [**setStudentName**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#setStudentName-java.lang.String-)(java.lang.String sName)  Setter Method for Student Name Throws Exception if the name of the Student contains special characters or numbers |
| Void | [**showDMC**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#showDMC--)()  This Method Displays the DMC of Student with the detailed view of every semester showing the Semester GPA and the CGPA overall. |
| java.lang.String | [**toString**](https://adnanmuhib.github.io/CGPA-Caclculator-UET/driver/UetGradeBook.html#toString--)()  This function Displays the Course Details in Tabular Form on a Dialogue box and always returns an Empty String |

# StudentDetails

**Constructor:**

StudentDetails()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

button-add action listener

database-

1. BufferedReader- to match individual student password

2. printwriter for copying that student information in another temp file

# StudentDetails

**Constructor:**

StudentDetails()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

menu-for design

menu Item-add action listener

database-

1. BufferedReader- to mach individual student info and set into a text field

# StudentCourseDetails

**Constructor:**

StudentCourseDetails()

//initcomponents

**Method**

Initcomponents()

//for all design adding

**Main Method**

public static void main(java.lang.String[] args)

**Parameters:**

args - the command line arguments

database-

1. BufferedReader- to match individual student course info
2. Set default table to show info

**Name: Mehjabin Johra**

**ID:011171037**

**Project Name: Student Management System**

**Date: 12-01-2019**