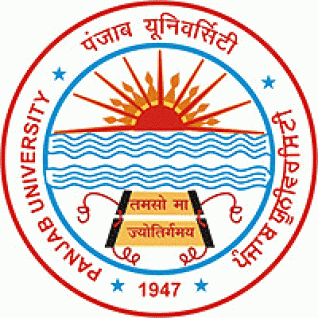
**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS**

**PANJAB UNIVERSITY SWAMI SARVANAND GIRI REGIONAL CENTRE**

**HOSHIARPUR**

****

**Project Report On**

**“Marks Evaluation System”**

**Submitted by:**

Shivangi Thakur

MCA

**CONTENTS**

* [Introduction](#Introduction)
* [Tools Used](#Tools_Used)
* [Designing the program](#DESIGN)
* [Data Flow Diagram](#DFD)
* [Implementation](#IMPLEMENTATION)
* [Interface](#INTERFACE)
* [Testing](#TESTING)
* [Future Scope](#FUTURE_SCOPE)
* [Bibliography](#BIBLIOGRAPHY)

**INTRODUCTION**

The process of teaching and learning in one or the other subjects of the curriculum always works for the realization of the stipulate learning objectives by following a well through program. A teacher strives hard for the achievement of the desired aims and objectives of teaching within the specific time period. Meanwhile, a teacher always assesses the students. Assessment is a mean to provide constant feedback to the learner to make the course effective.

Every year many students prepare the detailed project outline/synopsis in consultation with his/her Supervisor, identifies the topic of the Project and submit it to their internal guide and they evaluated students on the basis of the various interim evaluation components, contents of the report.

Keeping records manually of each student is a hectic and difficult task for the teacher/guide . This application will help a teacher/guide to maintain all the records together as well as provide a easy access through the information of a particular student within seconds.

Various object oriented programming concepts were used to complete this project such as:

* Classes
* Objects
* Polymorphism
* Abstraction
* Encapsulation
* Inheritance

TOOLS USED

* Language

Java (JDK-13.0.1)

* IDE

Eclipse IDE for Java Developers - 2020-03

Visual Studio Code 1.45.0

* Plug-in

Windows Builder

* Operating System

Windows 10 version 2004

DESIGNING THE PROGRAM

Based on user needs and detailed analysis of the system a software must be designed. It is the most crucial phase in the development of a system. In this phase, the process continues to move from **what** question to **how** question. The logical design is turned into a physical design.

The techniques used for describing the system design of our application is Data Flow Diagram (DFD).

Data Flow Diagram

Main Page

Settings

Change Deadlines

Change Files related

Write

Read

Read

Read & Write

Add Student

Check Delay

Display all Students

Database (File storing data related to student)

Search/Edit Student

The arrow here represent the flow of data(to/from the database).

IMPLEMENTATION

**Windows Builder :**

WindowBuilder makes it very easy to create Java GUI applications without spending a lot of time writing code. Use the WYSIWYG visual designer and layout tools to create simple forms to complex windows; the Java code will be generated for you. Easily add controls using drag-and-drop, add event handlers to your controls, change various properties of controls using a property editor, internationalize your app and much more. WindowBuilder is built as a plug-in to Eclipse IDE that we used to develop the program.

**Package used :**

The packages used in this program are :

* java.awt.\*;
* java.awt.event.\*;
* java.awt.print.\*;
* java.text.\*;
* java.util.\*;
* javax.swing.\*;
* java.time.\*;
* java.io.\*;

Each one of the above packages plays a vital role in the execution of the program and it provides the program with a variety of libraries to use.

**Classes Created :**

Classes in a program is the building blocks of the program that holds different methods and variables, vital for the execution of the program.

In this program two classes were created :-

1. **student** : This class is used to store information related to a student. The member variables of this class are: int roll, String name, int ext, int synop, int letter, int prog1, int prog2, int prog3.
2. **myarray**: This class is used to store the content loaded/copied from file in the disk. The copied content is saved in a ArrayList. The member variables of this class are: ArrayList<student> arraylist, ArrayList<String> filelist, String activefile, Preferences prefs.

**Methods Used :**

Methods form the functional part of a program. In this program, different methods have been created to handle different functionalities.

Methods used in ou program are :

* public static void readArrayList();
* public static void writeArrayList();
* public static ArrayList<student> returnlist();
* public static void add(student s);
* public static int searchByName(String name);
* public static int searchByRoll(int roll);
* public static void delete(int i);
* public static void replace(int index, student s);
* public String field(int i);
* public int getMarks(int choice);

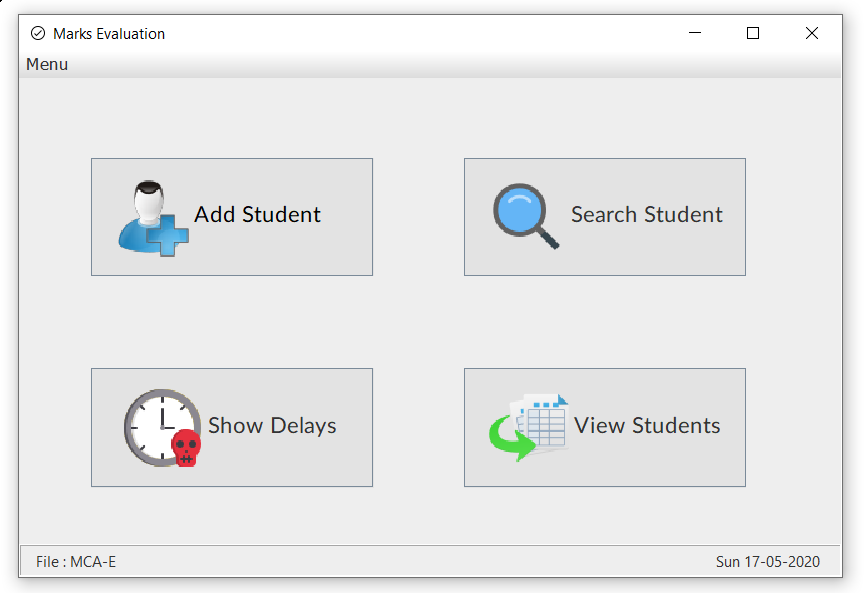
**Event Handling :**

Changing the state of an object is known as an event. For example, click on button, dragging mouse etc. The java.awt.event package provides many event classes and Listener interfaces for event handling. A lot of event handler user used throughout the program some of them are:

* actionPerformed
* mousePressed
* windowOpened
* windowClosed
* windowGainedFocus
* mouseEntered
* keyReleased

INTERFACE

This is main/index page of our application. This window provides many options through which you can access many functionality of application.



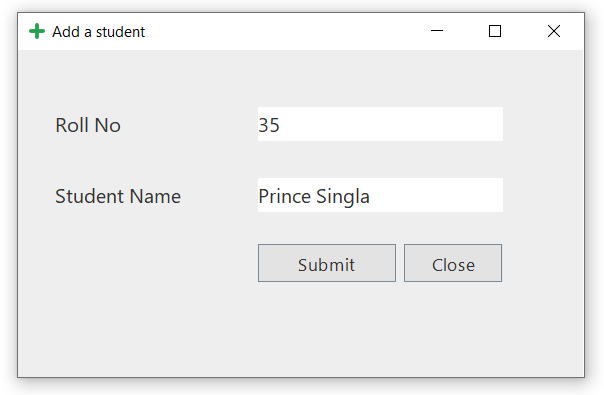
Date

File Name

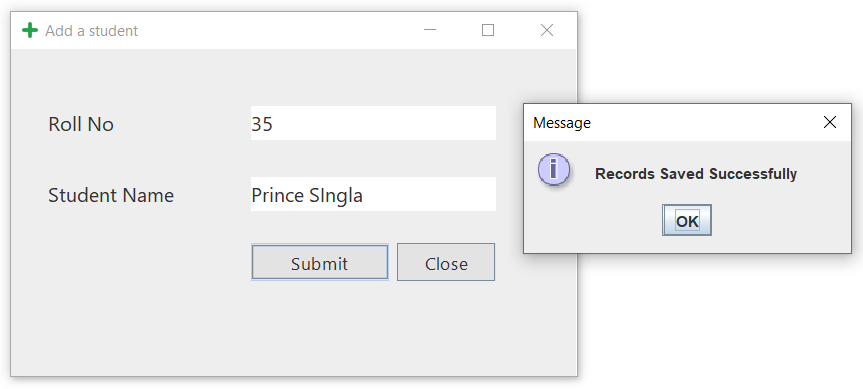
This front page also display filename currently being opened and today’s date extracted from the computer itself.

**ADD STUDENT**

Selecting add student will lead to this window from where you can add new student

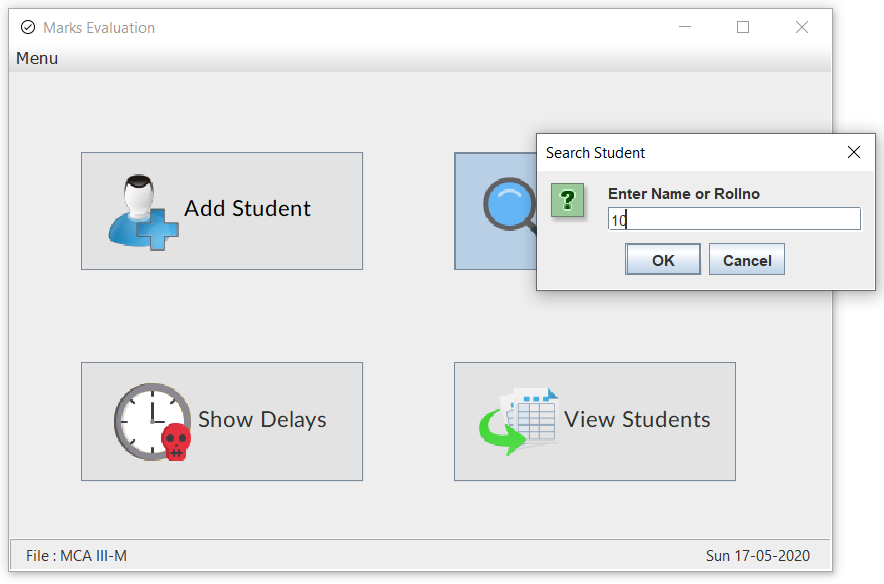


Click submit to save the record successfully

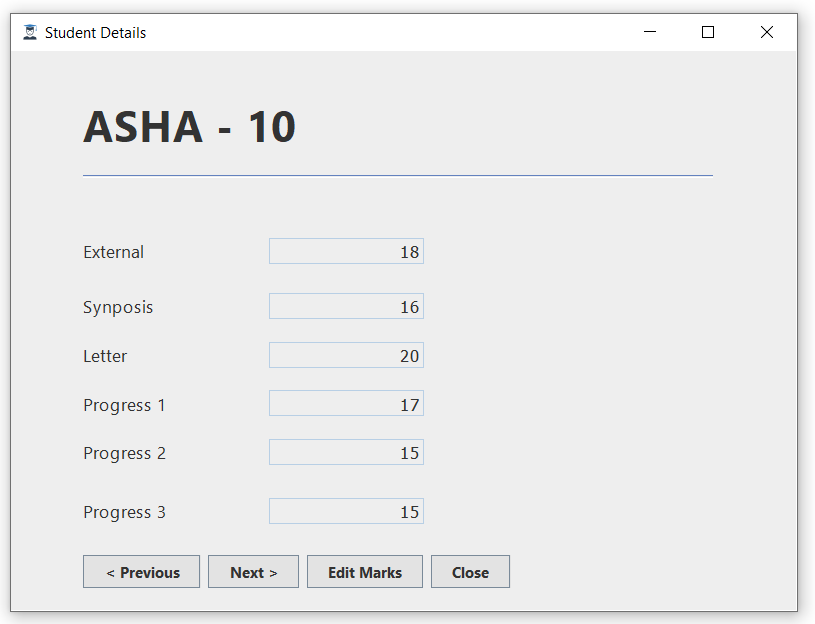


**Search Student**

A message box appears when you click search student. Here you can enter either a student’s Roll No or Name.



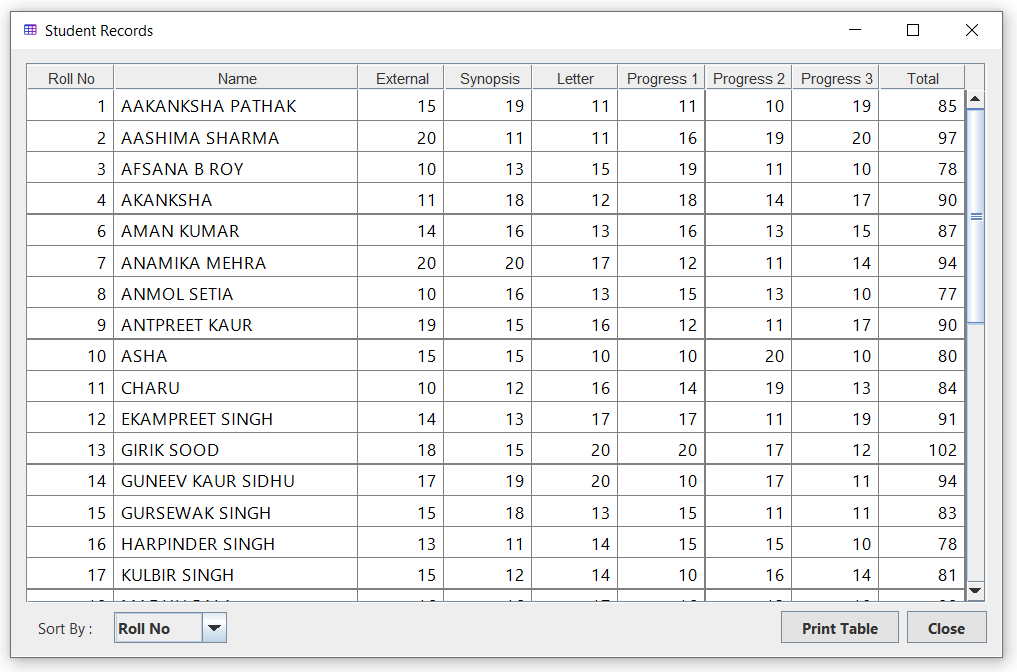
If found, a new window will appear with all the student details .



Here you can edit student marks and view previous or next record.

**View Students** :

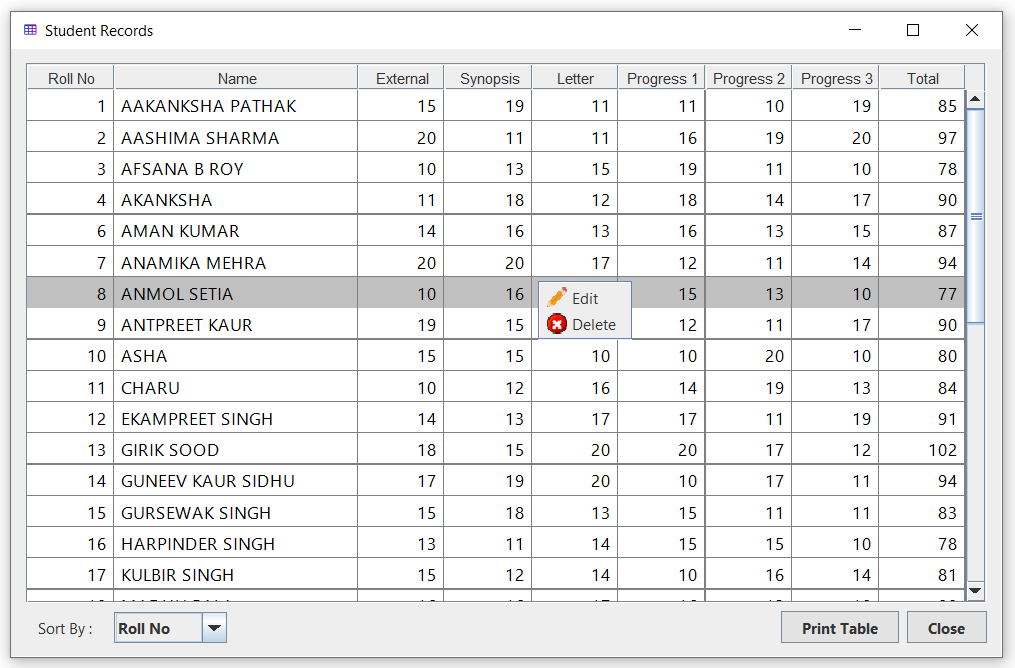
It will display all the students along with their roll no and marks in each assessment . A column calculating the total marks is also generated.



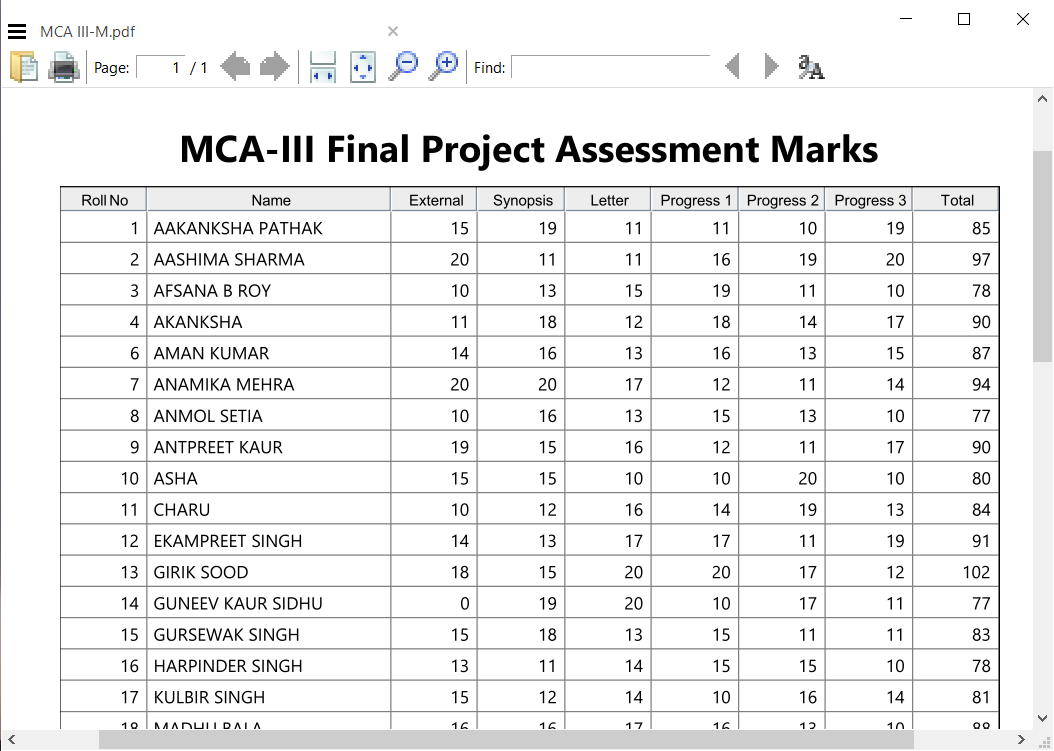
These records can be sorted by(Ascending Order) :

1. Roll No (Default)
2. Name
3. Total

You can also right click on any record in order to edit or delete that particular record.

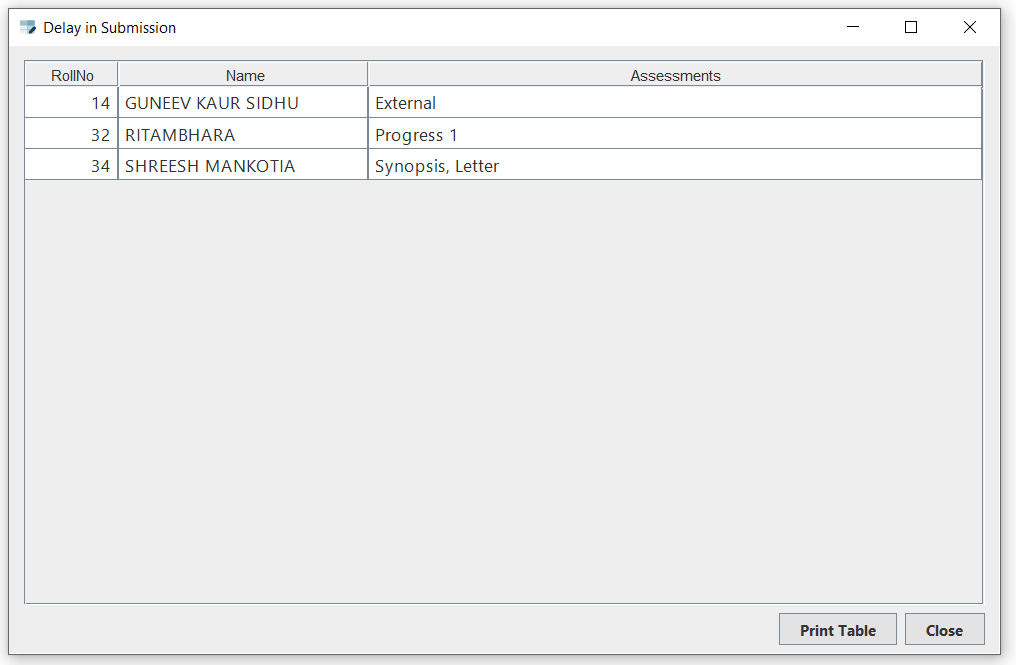


We can print the students record as a pdf file.



**Show Delays :**

This window will show all those students records who have missed any of the assessments deadline.



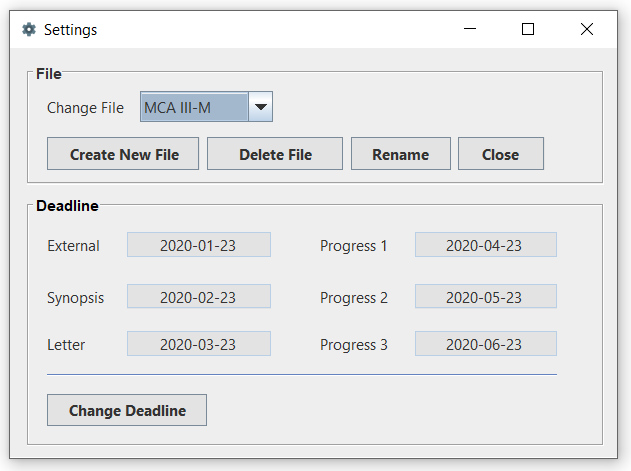
This also shows exactly in which assessment they have missed their deadlines. The deadlines of assessments can be changed anytime from settings.

You can also print table contents in pdf format.

**Settings :**

There are lot of options provided in settings page. You can do following things in settings:

1. Change current file being used.
2. Create a new file.
3. Delete an existing file.
4. Rename an existing file.
5. Change deadline of various assessments.



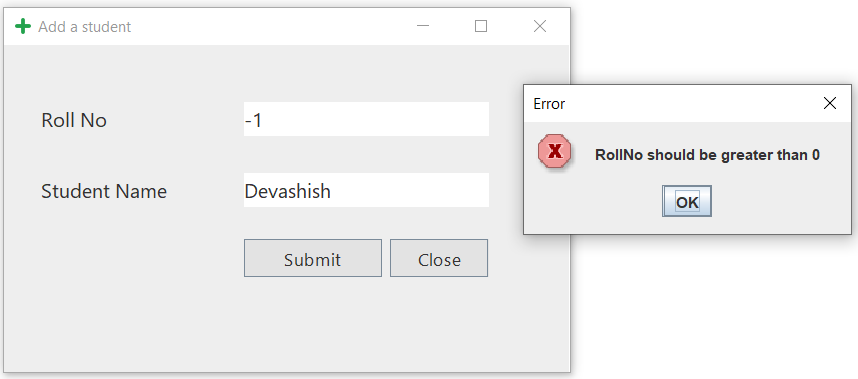
TESTING

The objective of the testing is to find and report as many bugs as possible to improve the integrity of the program. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements.

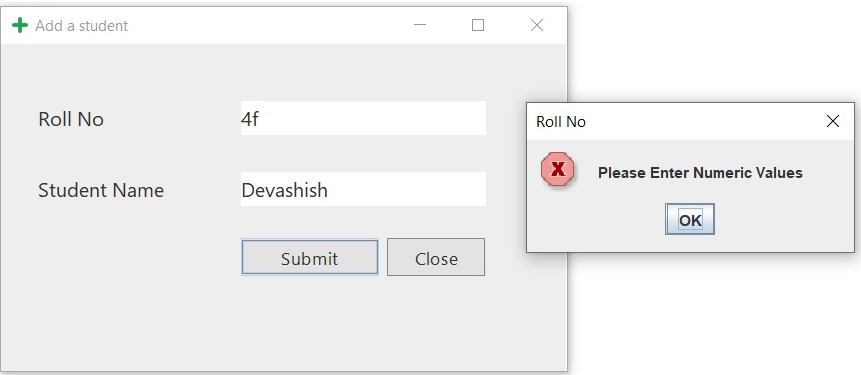
The Exception Handling in Java is one of the powerful mechanism to handle the runtime errors so that normal flow of the application can be maintained. Many exceptions were identified and handled during testing phase.

Following validations were added after encountering various exception in application:

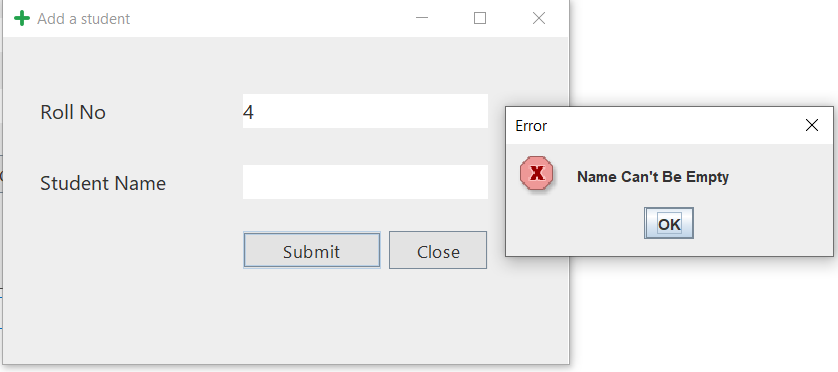
1. Roll no. of a student can’t be 0 or negative value.



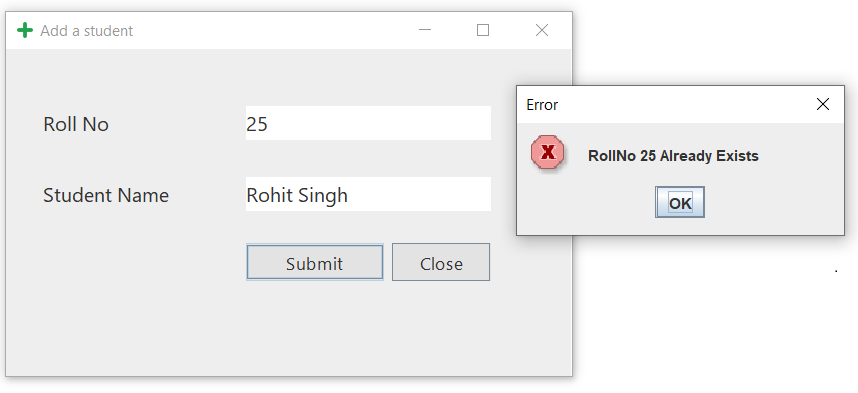
1. Rollno. can’t contain numeric values.



1. A name field can’t be empty.

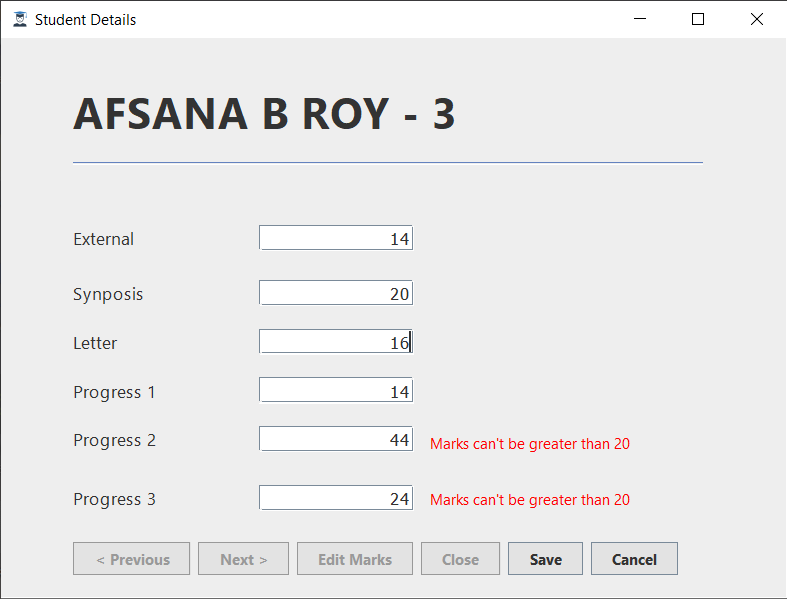


1. Two students can’t have same Rollno.

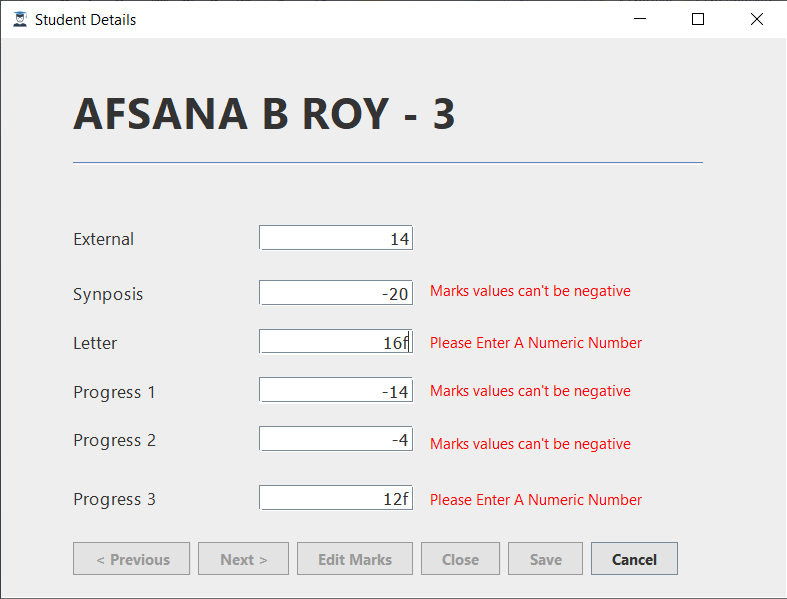


1. The marks of a student can’t be greater than 20.

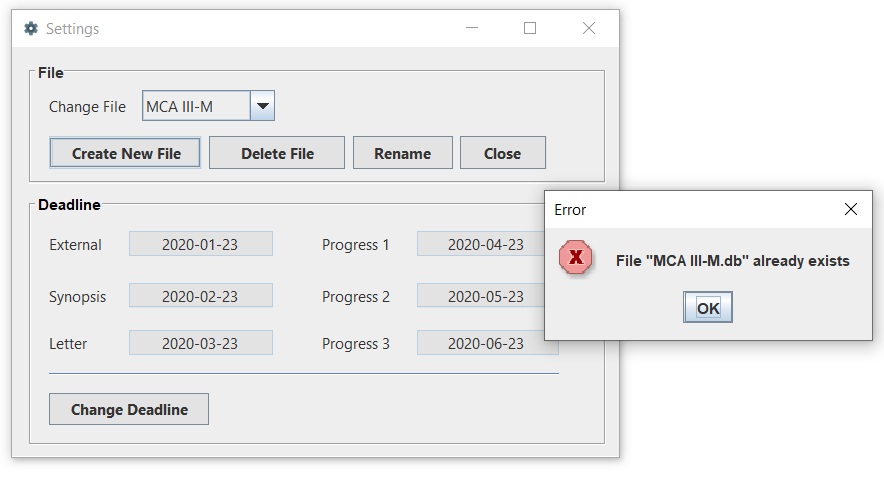
The static variable int MAX\_MARKS=20 declared in student class is the upper bound that these fields follow:



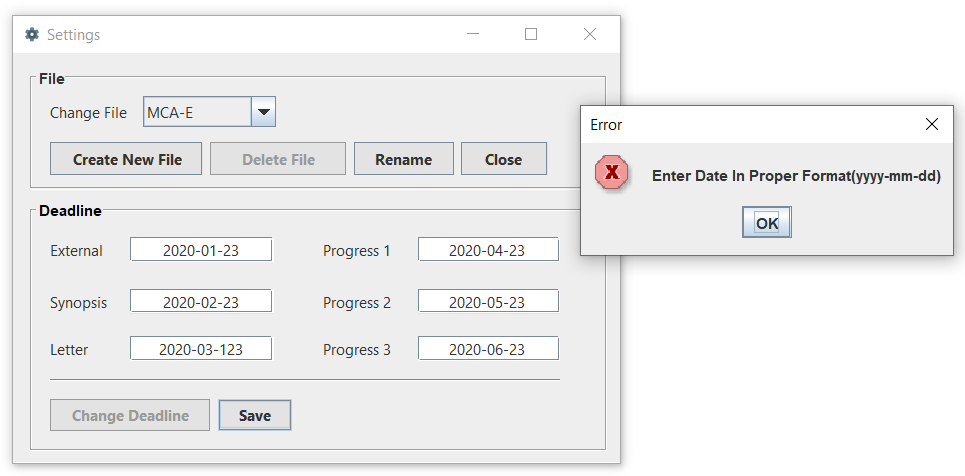
1. Marks can’t be negative and should contain numeric digits only.



1. If a file you’re trying to create already exists, will result in an error instead of overwriting the file.



1. Entering a date in wrong format will result in an error.



FUTURE SCOPE

* I would like to further improve my application by deploying my application to web so that user don’t need to install a separate app, they can browse the website for same functionality. Deploying my app to web will also mean that it will work on every machine independent of its architecture. All they need is a web browser.
* I also have plans to make my database online so that I don’t have to store the database offline in user machine, which will increase the data security.
* The data stored needs to be encrypted to increase security.

BIBLIOGRAPHY

* Programming with Java 5e (E Balaguruswami)
* Javapoint
* Tutorialspoint
* YouTube
* StackOverFlow