

Mehran University Of Engineering And Technology



Project Report

Data Structure And Algorithm

Submit To	Dr. Sammer Zia	
Sec	1	
Department	Computer System	
Project name:	Student management system using stack data structure	
Submitted by:	Group Member	
	Naveen khan	21CS059
	Isra samoo	21CS029
	Mahwish somroo	21CS047
	Hania Yasir	21CS017

Department of Computer Systems Engineering
Mehran University of Engineering and Technology, Jamshoro

Course: Data Structure and Algorithm Analysis (CS-251)

Instructor	Dr. Sammer Zai	Assignment Type	PBL based Project
Semester	4 th	Year	2 nd
Batch	21CS	Section	I
Submission Deadline		Assessment Score	05

Problem Description

Choose a real-world problem to construct a project related to any domain of your choice e.g.; education, entertainment, health care, industry, management, etc. The project should clearly define the objectives and reason for the selection of specific data structure(s) towards obtaining the problem's solution. Also provide GUI if applicable. Moreover, the project report should comprise of the following sections:

- 1- Problem Description.
- 2- Block Diagram or Flow Chart (to describe the working of complete project)
- 3- Implementation Details (should describe the selected data structure(s) along with the coding details).
- 4- Results (should describe the output of the project by testing various inputs).
- 5- Project Complexity. (describe the complexity (time/space complexity) of the data structure you selected for implementation of the overall project).

Evaluation of this project will be carried out by using the following five rubrics.

Assessment Rubrics

S#	Rubrics	Assessment					
		Unacceptable (0)	Poor (2)	Acceptable (5)	Adequate (8)	Proficient (10)	Total Score (10)
R1	Deeper understanding of the real-world problem.						
R2	Well defined content objectives.						
R3	Level of complexity to ensure that the students must work together.						
R4	Reasoned decisions and to defend them.						
R5	Technical Writing and Presentation.						
Cumulative Marks (out of 50)							
Total Marks out of 5 = Cumulative Marks/50 *5							

Implementation Of Student Management System Using Stack Data Structure

Aim of Selecting Stack Data Structure:

The fundamental aim of selecting the stack data structure for the Student Management System is its simplicity and effectiveness in certain operations, especially in situations where we require functionalities like 'undo' operations.

Objectives:

- ❖ **Fast Access to Recent Records** (nature of a stack (LIFO - Last In, First Out))
- ❖ **Simplified Data Management** (linear nature of a stack, certain operations become simpler)
- ❖ **Enhancing Understanding of Data Structures**(Implementing a real-world application)
- ❖ **Challenge and Innovation**
- ❖ **Undo Functionality**
- ❖ **Efficiency** (Access, addition, and removal of elements has time complexity is $O(1)$)
- ❖ **Space Optimization:**(don't need any additional pointers to perform the undo operation)

Introduction:

The Student Management System is developed to manage student records effectively. The system provides functionalities to insert student data, show records, search for a specific student, count students, delete student records, insert the record into a file, and perform undo operations. The undo functionality allows the system to revert the most recent operations, enabling a safety net for accidental entries or deletions.

The Student Management System is a comprehensive code specifically tailored to optimize the handling and organization of student data within educational institutions. Built with modern coding standards, the system is equipped to function seamlessly and provide user-friendly interfaces to streamline academic administrative tasks.

Methodology:

The architecture of our Student Management System is grounded in a straightforward and effective methodology.

❖ **Data Representation:**

We use a student structure that encapsulates essential details like name, id, qualifications, contact, and roll number. This allows for a structured and logical representation of each student.

❖ **Undo Mechanism:**

To offer flexibility in operations, we added an 'undoStack'. Every time we do something to the student data, it's saved in this stack. So, if we accidentally delete or change something, the undo button checks this stack and fixes our mistake.

❖ **Data Storage:**

For managing multiple student records, we've deployed an array of student structures. This approach ensures efficient storage and quick access.

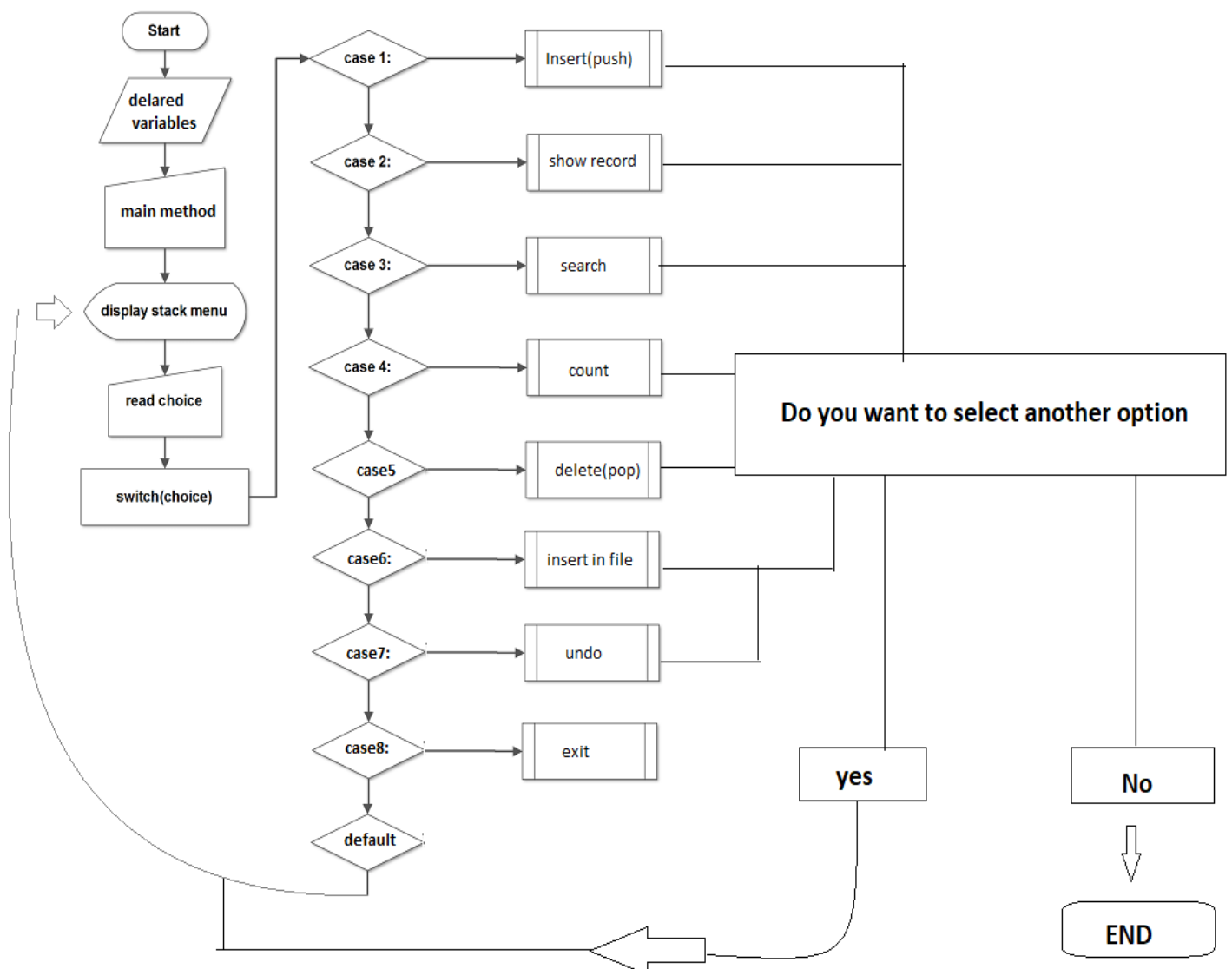
❖ Modular Design:

The system is divided into specific functions, each catering to a distinct operation. This modular design not only promotes clarity but also simplifies any debugging process

Key features:

- ❖ Insertion of student records.
- ❖ Display of all student records.
- ❖ Searching for a student using ID.
- ❖ Deletion of the most recent record.
- ❖ Count student records
- ❖ Option to write all records to a file.
- ❖ Undo operation
- ❖ Exit

Flow chart



Algorithm:

1. Initialization:

- ❖ Include necessary libraries.
- ❖ Declare global structures for student and action.
- ❖ Declare global arrays for storing students (s[]) and undo operations (undoStack[]).
- ❖ Set initial values for total_students, top, and undoTop.

2. Main Function:

- ❖ Display the main menu options.
- ❖ Loop to allow the user to perform operations repeatedly.
- ❖ Call list_opeation() function.

3. list_operation Function:

- ❖ Clear the screen.
- ❖ Display available options.
- ❖ Get user's choice.
- ❖ Use a switch case to determine the operation.
 - If 1, call insertion_students_records().
 - If 2, call show().
 - If 3, call Search().
 - If 4, call count_number_of_students().
 - If 5, call delt().
 - If 6, call insert_record_in_file().
 - If 7, call undo().
 - If 8, exit the program.
 - Otherwise, display an error message.
- ❖ Ask the user if they want to select another option. If yes, loop back. If no, exit.

4. Insertion_students_records Function:

Check if the limit (maximum_size) is reached. If yes, display error and return.

- ❖ Increment top.
- ❖ Prompt user to enter details for a new student.
- ❖ Store student details in s[top].
- ❖ Push this student to undoStack with "insert" operation.
- ❖ Increment total_students.

5. show Function:

- ❖ Check if total_students is 0. If yes, display a message.
- ❖ Display all student records from s[].

6. Search Function:

- ❖ Prompt user for a student ID.
- ❖ Iterate through s[] to find the student with the given ID.
- ❖ Display the student details if found. If not, show an error message.

7.count_number_of_students Function:

- ❖ Display the value of total_students.

8.delt Function:

- ❖ Check if top is -1, indicating an empty list. If yes, display error and return.
- ❖ Push the current top student to undoStack with "delete" operation.
- ❖ Decrement top.
- ❖ Decrement total_students.

9. insert_record_in_file Function:

- ❖ Open a file for appending.
- ❖ Iterate through s[] and write each student's data to the file.
- ❖ Close the file.

10 .pushToUndoStack Function:

- ❖ Check if undoTop exceeds maximum_size. If yes, display an error and return.
- ❖ Otherwise,Increment undoTop.
- ❖ Store student and operation in undoStack[undoTop].

11.undo Function:

- ❖ Check if undoTop is -1 indicating empty. If yes, display a message.
- ❖ If the latest operation is "insert":
 - ❖ Call delt() to remove the student.
 - Display a message indicating the undo of insertion.
- ❖ If the latest operation is "delete":
 - Increment top and restore the student from undoStack.
- ❖ Increment total_students.
 - Successfully perform undo,Display a message indicating the undo of deletion.(recover data)
- ❖ Decrement undoTop

Implementation of code

```
//-----header files-----
#include<iostream>
#include<conio.h>
#include<windows.h>
#include <string>
#include <fstream>

using namespace std;

//-----functions-----
void display();
void list_opeation ();
void insertion_students_records();
void show();
void Search();
void delt();
void insert_record_in_file();
void count_number_of_students();
void undo();

//-----creating Structure -----
struct student{                // this structure store the data of 1 student at a time
    string name,id,qualification;
    long contact;
    string roll_number;
};

struct action{                 //this structure store the data of 1 student and operator at a time
    student s;
    string operation;          // can be "insert" or "delete"
};

//-----declaretion of variable -----
const int maximum_size=100;

student s[maximum_size];      // we are creating array of structure data type "student"
```

```

// array:1

static int total_students=0;

static int top=-1;

// ****--variable for undo operation--****
//undo stack is a array whose 1 element stored the student of data and operator value
action undoStack[maximum_size];

int undoTop = -1;

void pushToUndoStack(student , string );

//****--main method--****

int main(){

    display();

    list_opeation();

}

//****--display method--****

void display(){

cout<<"\n\n\t****";

cout<<"\n\n\t\tSTUDENT MANAGMENT SYSTEM"<<endl;

cout<<"\n\t****";

cout<<"\n\n\t\tPress 1 to Insertion student recods "<<endl;

cout<<"\t\tPress 2 to show student recods"<<endl;

cout<<"\t\tPress 3 to search student recods"<<endl;

cout<<"\t\tPress 4 to count students"<<endl;

cout<<"\t\tPress 5 to delete student recods"<<endl;

cout<<"\t\tpress 6 to insert record in file"<<endl;

cout<<"\t\tPress 7 to undo operation"<<endl;

cout<<"\t\tPress 8 to exit"<<endl;

}

//****--display method--****

```



```

void list_opeation (){
char ch;
char c;          // for do while condition
do{
system("CLS");
display();          //display the list
cout<<"\n enter your choice of records :";
cin >>ch;
cout<<"\t\t\nplease wait a moment";
for(int i=0;i<6;i++)
{
    cout<<".";
    Sleep(500);
}
system("CLS");
switch(ch){
case '1':
    insertion_students_records();
    break;
case '2':
    show();
    break;
case '3':
    Search();
    break;
case '4':
    count_number_of_students();
    break;
case '5':
    delt();
    break;
}
}
}

```

```

        case '6':
            insert_record_in_file();

            break;

        case '7':
            undo();

            break;

        case '8':
            exit(0);

            break;

        default:
            cout<<"\aInvalid Input"<<endl;

            break;

    }

    cout<<"Do you Want to select another..(y/n)!"<<endl;
    cin>>c;

    }while(c=='y' || c=='Y');

    cout<<" byee Nice to meet you...";

    exit(0);

}

```

[illegible]

```

else
{
    top++;
    cout<<"\n \tenter data of student "<<total_students+1<<"\n\n ";
    cout << "\tEnter student name: ";
    cin>>s[top].name;
    cout<<"\n\tEnter student id: ";
    cin>>s[top].id;
    cout<<"\n\tEnter student qualification: ";
    cin>>s[top].qualification;
    cout<<"\n\tEnter contact: ";
    cin>>s[top].contact;
    cout<<"\n\tEnter Roll : ";
    cin>>s[top].roll_number;

    //insert data in undo stack
    // this is used for undo operation.
    pushToUndoStack(s[top], "insert");
    total_students=total_students+1;
}
}

//--*-*-*-*-*-*-*-*-*-*-*--show records of students --*-*-*-*-*-*-*-*-*-*-*

void show(){
    //All data show which is entered
    cout<<"\n\n\t--*-*-*-*-*-*-*-*-*-*-*";
    cout<<"\n\n\t\tshow Data of students"<<endl;
    cout<<"\n\n\t--*-*-*-*-*-*-*-*-*-*-*\n";
    int count=1;
    if(total_students!=0){
        for(int i=top;i>=0;i--)        // show data from top most position
        {
            cout<<"\n\tData of STUDENT "<<count<<endl;

```

[illegible]

```
void Search(){  
    //search the id of specific student  
    cout<<"\n\n\t--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*";  
    cout<<"\n\n\t\tSearch Data of student"<<endl;  
    cout<<"\n\n\t--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*\n";  
    if(total_students!=0)        //IF(1)  
    {  
        string id;  
        cout<<"Enter id of student which you want to search"<<endl;  
        cin>>id;  
        for(int i=0;i<=top;i++){  
            if(s[i].id==id){        //IF(2)  
                cout<<"Name: "<<s[i].name<<endl;  
                cout<<"ID: "<<s[i].id<<endl;  
                cout<<"qualification: "<<s[i].qualification<<endl;  
                cout<<"Contact: "<<s[i].contact<<endl;  
                cout<<"Roll number: "<<s[i].roll_number<<endl;  
                break;  
            }  
            //end of IF(2) statement  
        }  
    }  
}
```

```

if(i==total_students-1){
    cout<<"No such record found"<<endl;
}
} //end of loop
} //END of OUTER IF(1)
else{
    cout<<"No data is entered"<<endl;
}
}

//--*--*--*--*--*--*--*--*--*--*--*--*--*--countT records of students --*--*--*--*--*--*--*--*--*--*--

void count_number_of_students(){
system("CLS");
cout<<"\n\n\t--*--*--*--*--*--*--*--*--*--*--*--*--*--";
cout<<"\n\n\t\t count number of students";
cout<<"\n\n\t--*--*--*--*--*--*--*--*--*--*--*--*--*\n";
if(total_students==0){
    cout<< "\n\t\t sorry\n record was not entered\n\n";
}
else
{
    cout<<"\n\t\t total number of student record you entered is:"<<total_students<<endl;
}
}

//--*--*--*--*--*--*--*--*--*--*--*--*--*--delete records of top recent student --*--*--*--*--*--*--*--*--*--*--

void delt(){
cout<<"\n\n\t--*--*--*--*--*--*--*--*--*--*--*--*--*--";
cout<<"\n\n\t\t DELETE STUDENT RECORD"<<endl;
cout<<"\n\n\t--*--*--*--*--*--*--*--*--*--*--*--*--*--";
cout<<"\n\n delete record of recent student\n";
cout<<"\n\n\t\t please wait a moment";
for(int i=0;i<7;i++)

```

```
{
cout<<".";
Sleep(500);
}
if(top== -1){
cout<<"\n\tdeletion is not possible\n\t";
}
else{
    pushToUndoStack(s[top], "delete"); // Before deleting, save to undoStack
top--;
total_students--;
}
cout<<"\n\nrecord successfully delete!...";
}
//-----write records of student in file-----

void insert_record_in_file(){
{
system("CLS");

cout<<"\n\n\t-----";
cout<<"\n\n\t\tinsert record"<<endl;
cout<<"\n\n\t-----\n";
cout<<"\n\n\n\tinsert records of studens...";
cout<<"\n\n\n\tplease wait a mint";
}

for(int i=0;i<6;i++){
cout<<".";
Sleep(700);
}

ofstream file("items.file",ios::app);

    // Check if the file has been opened correctly.
if(!file) {
    cout << "Error opening file for writing!" << endl;
```

```

        return;
    }

    file << "data of Student: " << "\n";
    for(int i=top; i>=0; i--) {

        file << "Name: " << s[i].name << "\n";
        file << "ID: " << s[i].id << "\n";
        file << "Qualification: " << s[i].qualification << "\n";
        file << "Contact: " << s[i].contact << "\n";
        file << "ROLL NUMBER: " << s[i].roll_number << "\n";
        file << "-----" << "\n"; // Separator for readability
    }

    file.close();

    cout << "\n\t\tData successfully written to file." << endl;
}

//----- perform undo operation on data -----
void undo(){
    cout<<"\n\n\t-----";
    cout<<"\n\n\t\tundo operation ";
    cout<<"\n\n\t-----\n\n";
    if(undoTop == -1){
        cout << "\n\t Nothing to undo!";
        return;
    }
    if(undoStack[undoTop].operation == "insert"){
        delt();
        cout << "\n\tLatest insertion has been undone!";
    }
    else if(undoStack[undoTop].operation == "delete"){
        top++;
        s[top] = undoStack[undoTop].s;
        total_students++;
    }
}

```

[illegible]

OUTPUT

It is our first screen when we execute our program initially no data is entered when we show the data by pressing 2



```
D:\program\with undo.exe

-----*
show Data of students
-----*

No data is entered
Do you Want to select another option..(y/n)!_
```

Now we are inserting some data in order to perform operation on it



```
D:\program\with undo.exe

-----*
STUDENT MANAGMENT SYSTEM
-----*

Press 1 to Insertion student recods
Press 2 to show student recods
Press 3 to search student recods
Press 4 to count students
Press 5 to delete student recods
press 6 to insert record in file
Press 7 to undo operation
Press 8 to exit

enter your choice of records :_
```

here you see that we successfully entered data of first student then it ask to the use that you perform more operation on data or not if yes then it show the list of operations

```
D:\program\with undo.exe
```

```
-----*-----  
Enter Insert student records  
  
-----*-----  
enter data of student 1  
Enter student name: NAVEEN  
Enter student id: 12  
Enter student qualification: INTER  
Enter contact: 123455678  
Enter Roll : 21CS059  
Do you Want to select another option..(y/n)!Y_
```

Now we select option 2 to insure that our data is entered successfully

```
D:\program\with undo.exe
```

```
-----*-----*-----*-----*-----*-----*-----*-----*-----*  
STUDENT MANAGMENT SYSTEM  
-----*-----*-----*-----*-----*-----*-----*-----*-----*  
  
Press 1 to Insertion student recods  
Press 2 to show student recods  
Press 3 to search student recods  
Press 4 to count students  
Press 5 to delete student recods  
press 6 to insert record in file  
Press 7 to undo operation  
Press 8 to exit  
  
enter your choice of records :2
```

Here is the result .. now we select another option to perform another operation

```
D:\program\with undo.exe  
  
-----  
show Data of students  
  
-----  
Data of STUDENT 1  
Name: NAVEEN  
ID: 12  
Qualification: INTER  
Contact: 123456  
Roll number: 21CS059  
Do you Want to select another option..(y/n)!_
```

Now we insert another data of student

```
D:\program\with undo.exe
```

```
--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
```

```
STUDENT MANAGMENT SYSTEM
```

```
--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
```

```
Press 1 to Insertion student recods  
Press 2 to show student recods  
Press 3 to search student recods  
Press 4 to count students  
Press 5 to delete student recods  
press 6 to insert record in file  
Press 7 to undo operation  
Press 8 to exit
```

```
enter your choice of records :1
```

```
please wait a moment..._
```

```
D:\program\with undo.exe

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
Enter Insert student records

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*

enter data of student 2

Enter student name: HINA

Enter student id: 123

Enter student qualification: MBA

Enter contact: 123456

Enter Roll : 21MA21
Do you Want to select another option..(y/n)!
```

Here you see that our second data is also entered now we want to perform more operation on data so, we select “y”. As you seen below we select option 3 to search the student by his/her id here the data is found ...now we select another operation.

```
D:\program\with undo.exe

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*

Search Data of student

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*

Enter id of student which you want to search
123
Name: HINA
ID: 123
qualification: MBA
Contact: 123456
Roll number: 21MA21
Do you Want to select another option..(y/n)!
```

```
D:\program\with undo.exe
```

```
-----*-----  
STUDENT MANAGMENT SYSTEM  
-----*-----  
  
Press 1 to Insertion student recods  
Press 2 to show student recods  
Press 3 to search student recods  
Press 4 to count students  
Press 5 to delete student recods  
press 6 to insert record in file  
Press 7 to undo operation  
Press 8 to exit  
  
enter your choice of records :4  
  
please wait a moment.._
```

A screenshot of a Windows command prompt window titled "D:\program\with undo.exe". The window has a dark background with white text. It displays the output of a C++ program. At the top, there are two horizontal dashed lines made of asterisks. Below them, the text "count number of students" is centered. This is followed by another set of two horizontal dashed lines made of asterisks. Then, the text "total number of student record you entered is:2" appears. Finally, the prompt "Do you Want to select another option..(y/n)!Y_" is shown, with the character 'Y' having been entered and the cursor positioned after it.

```
D:\program\with undo.exe  
  
_____*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_  
  
        count number of students  
  
_____*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_  
  
    total number of student record you entered is:2  
Do you Want to select another option..(y/n)!Y_
```

now,we want to select more option to perform operation

```
D:\program\with undo.exe  
  
--*-*-*-*-*  
STUDENT MANAGMENT SYSTEM  
--*-*-*-*-*  
  
Press 1 to Insertion student recods  
Press 2 to show student recods  
Press 3 to search student recods  
Press 4 to count students  
press 5 to delete student recods  
Press 6 to insert record in file  
Press 7 to undo operation  
Press 8 to exit  
  
enter your choice of records :5  
  
please wait a moment..
```

we know that we insert data of two student, now we perform deletion operation on data so, the data is deleted which is entered in last here we see that it show the data of s1 and data of s2 is deleted because stack work on last in first out so it delete the data which insert recently. Result shown in below

```

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
      show Data of students

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*

Data of STUDENT 1

Name: NAVEEN

ID: 12

Qualification: INTER

Contact: 123456

      Roll number: 21CS059
Do you Want to select another option..(y/n)!_

```

Now I perform undo operation:

```
D:\program\with undo.exe
```

```
--**--**--**--**--**--**--**--**--**--**--**--**--*
```

```
STUDENT MANAGMENT SYSTEM
```

```
--**--**--**--**--**--**--**--**--**--**--**--**--*
```

```
Press 1 to Insertion student recods  
Press 2 to show student records  
Press 3 to search student recods  
Press 4 to count students  
Press 5 to delete student recods  
press 6 to insert record in file  
Press 7 to undo operation  
Press 8 to exit
```

```
enter your choice of records :7
```

```
please wait a moment..._
```

successfully recover data:

```
D:\program\with undo.exe
```

```
--*_--*_--*_--*_--*_--*_--*_--*_--*_--*_--*_--*_--*
```

```
undo operation
```

```
--*_--*_--*_--*_--*_--*_--*_--*_--*_--*_--*_--*_--*
```

```
Latest deletion has been undone!Do you Want to select another option..(y/n)!y_
```

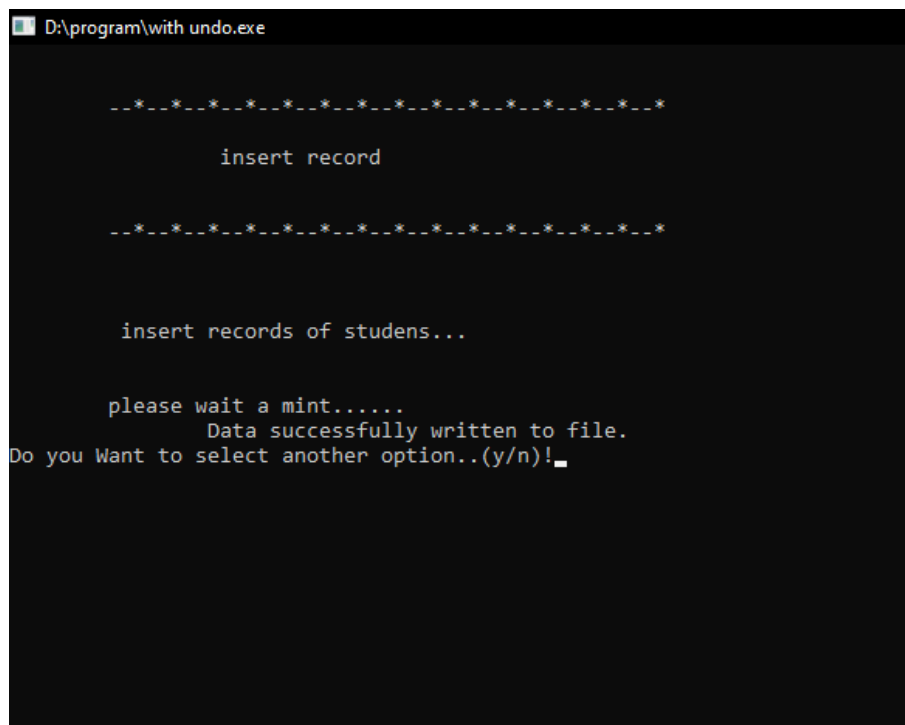
now we show that our data is recover or not?

```
D:\program\with undo.exe  
  
show Data of students  
  
--*-*-*-*-*-*-*-*-*-*  
  
Data of STUDENT 1  
Name: HINA  
ID: 123  
Qualification: MBA  
Contact: 21  
Roll number: MB  
  
Data of STUDENT 2  
Name: NAVEEN  
ID: 12  
Qualification: INTER  
Contact: 1234556  
Roll number: 21CS059  
Do you Want to select another option..(y/n)!Y
```

As shown in above we are successfully recover our data. Here noted that when we show the data to display it show the data first, which entered at last Now, we insert data in file

Initially our file is empty

as shown in below our data is successfully entered in file



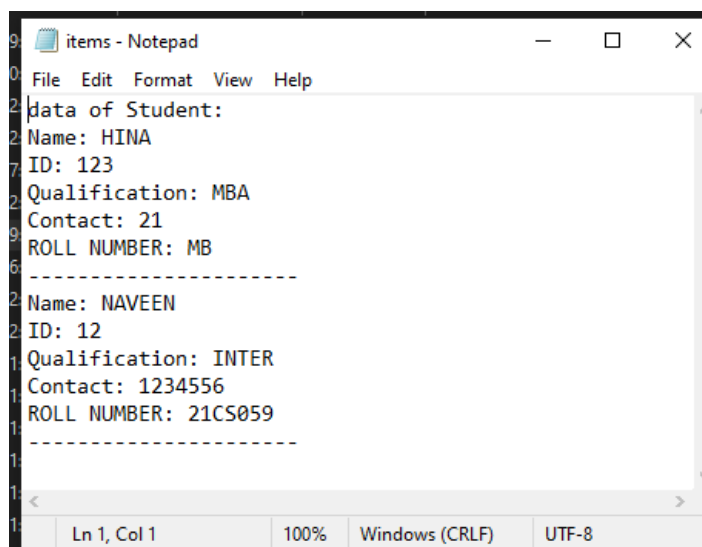
```
D:\program\with undo.exe

--*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
      insert record
--*-*-*-*-*-*-*-*-*-*-*-*-*-*

      insert records of students...

      please wait a mint.....
              Data successfully written to file.
Do you Want to select another option..(y/n)!_
```

now check in file



```
items - Notepad
File Edit Format View Help
data of Student:
Name: HINA
ID: 123
Qualification: MBA
Contact: 21
ROLL NUMBER: MB
-----
Name: NAVEEN
ID: 12
Qualification: INTER
Contact: 1234556
ROLL NUMBER: 21CS059
-----
Ln 1, Col 1    100%    Windows (CRLF)    UTF-8
```

The diagram illustrates the Undo operation in a database system. It shows a sequence of operations:

- An initial empty box.
- An **INSERT** operation (indicated by a right arrow) adding "Data of student1".
- An **UNDO** operation (indicated by a left arrow) removing "Data of student1".
- A second **INSERT** operation (indicated by a right arrow) adding "Data of student2".
- A second **UNDO** operation (indicated by a left arrow) removing "Data of student2".
- A final **UNDO** operation (indicated by a right arrow) leading to a table with two rows: "STUDENT2" and "STUDENT1".
- A **DELETE** operation (indicated by a double-headed vertical arrow) removing the "STUDENT2" row, leaving only "STUDENT1" in the table.

```
D:\program\with undo.exe  
  
-----  
show Data of students  
  
-----  
Data of STUDENT 1  
Name: HINA  
ID: 123  
Qualification: MBA  
Contact: 21  
Roll number: MB  
Data of STUDENT 2  
Name: NAVEEN  
ID: 12  
Qualification: INTER  
Contact: 1234556  
Roll number: 21CS059  
Do you Want to select another option..(y/n)!_
```

Now we perform undo operation again, then data of student 2 is delete

```

D:\program\with undo.exe

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
undo operation
--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
DELETE STUDENT RECORD
--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*

delete record of recent student

please wait a moment.....

record successfully delete!...
Latest insertion has been undone!Do you Want to select another option..(y/n)!

```

Then we show result:

```
D:\program\with undo.exe  
  
-----*-----  
show Data of students  
  
-----*-----  
Data of STUDENT 1  
Name: NAVEEN  
ID: 12  
Qualification: INTER  
Contact: 1234556  
Roll number: 21CS059  
Do you Want to select another option..(y/n)!_
```

Now again we perform undo operation the data of student 1 is also deleted and our stack become empty

```
D:\program\with undo.exe

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
undo operation
--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
DELETE STUDENT RECORD
--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*

delete record of recent student

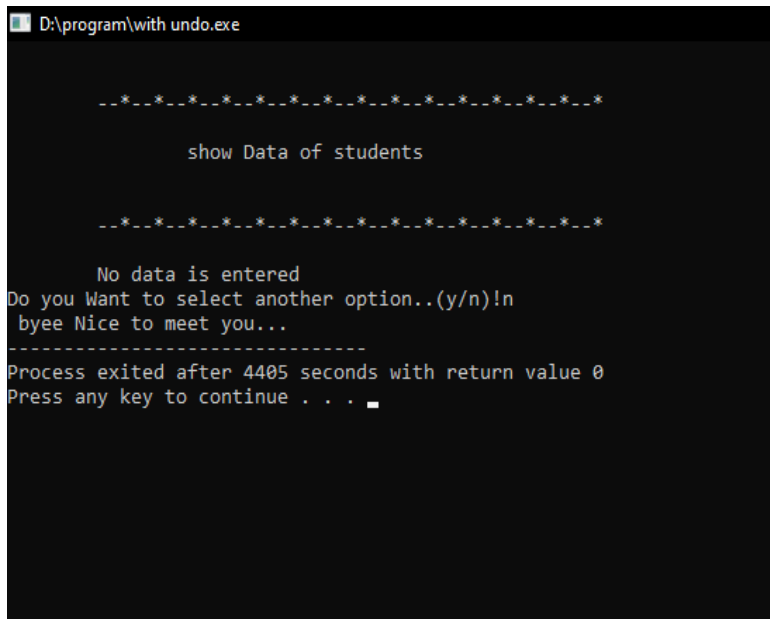
please wait a moment.....
```

```
D:\program\with undo.exe

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
show Data of students
--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*

No data is entered
Do you Want to select another option..(y/n)!
```

Now our stack is empty..if we enter 'n' which means we want to exit our program or terminate our program



```
D:\program\with undo.exe

--*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
show Data of students

--*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

No data is entered
Do you Want to select another option..(y/n)!n
bye Nice to meet you...
-----
Process exited after 4405 seconds with return value 0
Press any key to continue . . .
```

Conclusion

Our Student Management System embodies the practical implementation of stack data structures in addressing real-world needs. Leveraging the core concepts of stacks, our system not only organizes student data effectively records but also to allow undo operations effectively This feature is invaluable in many applications where users may need to reverse their last input or action.

The choice of stack makes certain operations swift and efficient. Every operation like inseting(push), deleting(pop), undo operation all has complexity of $O(1)$,by the use of stack we donot need of shifting algoithum for insertion or deletion like array

The incorporation of the undo feature in this management system showcases how theoretical concepts can be seamlessly integrated into practical applications to solve real problems.

Additionally, the system's ability to handle files allows for persistent storage of student data, making it more robust and user-friendly in real-world applications.