A

Project Report on

STUDENT RECORD MANAGEMENT SYSTEM

Submitted to the partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Divyansh Mathur (RA2211003030070)

Aaryman Misra (RA22110030300118)

Supervised by:

Dr. Oshin Sharma (Associate Professor)

SRM Institute of Science and Technology

Delhi NCR Campus, Modinagar, Ghaziabar(U.P) – 201204

OCTOBER 2023

BONAFIDE CERTIFICATE

This is to certify that the project entitled “Student Record Management System”, which is submitted by Divyansh Mathur(RA2211003030070) and Aaryman Misra (RA22110030300118) in the partial fulfilmentof the requirement for the award of degree B.Tech (CSE) of SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, Ghaziabad is a record of the candidate own work carried out by them under my own supervision.

……………………….

(Signature)

Dr. Oshin Sharma (Associate Professor)

ACKNOWLEDGEMENT

It gives us immense pleasure to express our deep sense of gratitude to Dr. Oshin Sharma, whose constant encouragement and valuable guidance have been the source of inspiration in the completion of this work. Without her sustained interest and sound counsel, this work would have not gained the present form.

We are equally thankful to all teaching faculty and ever willing supporting staff members and students of SRM IST Delhi NCR Campus Ghaziabad UP.

We are also thankful to my family for all their love and affection that helped us to fare well in my task. We thank all those who directly or indirectly helped us during this work.

Place: SRM UNIVERSITY NCR CAMPUS

Divyansh Mathur (070)
Aaryman Misra(118)

INDEX

INTRODUCTION

Abstract

This project presents a Sports Center Management System in C, utilizing data structures. It allows member management with insertion, deletion, display, and search capabilities. The system features a linked list for member storage and a stack for undo operations. This project showcases the practical implementation of data structures and algorithms in C for real-world applications.

SRMSs offer a number of benefits to educational institutions, including:

1) Improved efficiency and accuracy: SRMSs can automate many of the manual tasks involved in

student record keeping, such as data entry and reporting. This can save time and reduce errors.

2) Increased security and privacy: SRMSs can help to protect student data from unauthorized access and disclosure. They can also help to comply with data privacy regulations.

3) Enhanced communication and collaboration: SRMSs can facilitate communication and

collaboration between students, parents, teachers, and administrators. For example, students can use

SRMSs to access their grades, attendance records, and other important information. Parents can use

SRMSs to monitor their children's progress and communicate with teachers. Teachers can use

SRMSs to track student performance and provide feedback.

Significance of the Project

The Sports Center Management System project holds significant value as it demonstrates the practical application of data structures and algorithms in C for efficient member management. By utilizing linked lists and stacks, it showcases the robustness and flexibility of these structures in real-world scenarios. The project offers valuable insights into the development of management systems, enhancing organizational efficiency and user experience. It underscores the importance of sound data structure choices and serves as a foundation for more complex software solutions, contributing to the field of software engineering and data structure implementations.

The operations that can be performed using the Student record System are:

1) **Display Members**

2) **Insert Member**

3) **Delete Member**

4) **Search Member**

5) **Undo Add Member**

1. Objectives

The objectives of a Student Record System can be summarized as follows:

1. Efficient Data Management:To create a centralized system for managing student data, including personal information, academic records, and administrative details, to streamline data management and reduce redundancy.

2. Accuracy and Integrity: Ensure the accuracy and integrity of student records by minimizing data entry errors and maintaining a reliable database for academic and administrative purposes.

3. Accessibility and Retrieval: Facilitate easy and quick access to student records, allowing authorized users to retrieve, view, and update student information as needed.

4. Data Security: Implement robust data security measures to protect student records from unauthorized access, ensuring the confidentiality of sensitive information.

5. Automation: Automate routine administrative tasks such as grade calculations, attendance tracking, and report generation to enhance operational efficiency.

6. Academic Monitoring: Enable continuous monitoring of student academic progress, allowing faculty and administrators to assess student performance and make data-driven decisions.

2. SystemDesign

Switch Case in C/C++ Problem Statement: Write a program to build a simple

Software for Sports Center Management System which can perform the

following operations:

1) Store the Member ID

2) Store the member name .

.

Data Structures

Linked lists: Linked lists can be used to store member records in a linear order.

This allows for efficient insertion, deletion, and searching of member records.

Stack : Stack is a data structure that follows the LIFO (Last In First Out) principle. This means that the last element added to the stack is the first element to be removed. Stacks can be used to undo the last member that was added to the sports center management system

Databases: Databases can be used to store student records in a structured format. This allows for complex queries to be performed on the data, as well as for efficient storage and retrieval of large amounts of data.

Programming Languages and Tools

The project is implemented in C and uses standard libraries for input/ output (stdio.h) and dynamic memory allocation (stdlib.h).

3. Interactions:

- User Input

  Users interact with the system by providing input through the console interface. They can select options, enter data, and make choices based on the displayed menu.

- Menu Selection:

  Users interact with the system by choosing menu options to perform specific functions, such as displaying members, inserting a member, deleting a member, searching for a member, undoing the last addition, or exiting the program.

- Data Entry:

  Users interact with the system by entering data, such as member IDs and names, when inserting a new member. Data entry is essential for adding and updating member records.

- Data Display:

  Users can interact with the system to view data, such as the list of members. This interaction allows users to access and review member records.

- Search Interaction:

  Users can initiate a search for a member by providing a member ID, and the system interacts with the data structure (linked list) to find and display information about the member.

- Deletion Interaction:

  Users can interact with the system to delete a member by providing the member's ID. The system interacts with the data structure to remove the member's record.

- Undo Operation:

  Users can trigger the undo operation to reverse the last member addition. This interaction allows users to correct mistakes or make changes to the data.

- Error Handling:

The system interacts with users by providing error messages when input is incorrect or when specific operations cannot be performed due to constraints, such as duplicate member IDs.

- Loop for Continuous Interaction:

  The program is structured within a loop that continuously interacts with the user. It keeps the program running and allows users to perform multiple actions within the same session.

- Program Termination:

  Users can interact with the system to exit the program when they have completed their tasks. This interaction ensures that the program terminates cleanly.

- Data Structure Operations:

  The system interacts with the data structures (linked list and stack) to perform operations like adding, deleting, and undoing member records. These interactions ensure the integrity and management of member data.

4. Implementation Details

The implementation details of a Sports Center Management System would involve technical aspects and the use of programming languages, data structures, and databases to build the system. The key implementation details are:

1. Programming Language: Choose a programming language for system development. In this case, C was used.

2. Data Structures: Implement data structures to manage member records and undo operations. In the provided code, linked lists are used for member records, and a stack is used for undo operations.

3. Data Storage: Use data storage mechanisms, such as arrays or databases, to store and manage member information. In the code, a linked list structure is employed to store member records.

4. User Interface: Develop a user interface to interact with the system. The code includes a console-based interface that allows users to input commands and view results.

5. Authentication and Authorization: Implement user authentication and access control mechanisms to secure the system. In the code, user roles are not explicitly implemented, but access control can be added for a more comprehensive system.

6. Input Validation: Include input validation to ensure data accuracy and prevent errors. The code includes basic input validation for member ID and name.

7. Error Handling: Implement error handling to provide meaningful error messages and prevent system crashes. The code contains error messages for certain scenarios.

8. Database Integration: If using a database, establish a connection to the database system to store and retrieve member information. The code doesn't use a database, but the implementation can be extended to include one.

9. Report Generation: Develop mechanisms for generating reports and analysis. In the code, basic report generation is not included but can be added for more advanced functionality.

Functionality

Add Member :

Functionality: This option allows the user to add a new student record to the system.
Data Structure Usage: When a new member is added, a new node (a struct) is dynamically allocated to store the member's information. This new node is added to the beginning of the linked list, which represents the list of member records. The newly added member's informationis stored in this node.

Display Member:

Functionality: This option displays all the member records stored in the system.
Data Structure Usage: It traverses the linked list and prints the details of each student. The linked list serves as a data structure for storing and organizing the member records.

Search Member :

Functionality: Users can search for a member by providing the member ID.
Data Structure Usage: It traverses the linked list and checks if the name entered matches theI ID of any student. If a match is found, the member's details are displayed.

Delete Member:

Functionality: Users can delete a member record by entering the members ID.
Data Structure Usage: The program searches the linked list for a member with the specified member and deletes that member node. The deleted member's information is pushed onto the undo stack before deletion.

Undo Last Operation:

Functionality: This option allows the user to undo the last operation performed in the system.
Data Structure Usage: The undo stack (implemented as a stack data structure) is used to store the details of the last operation. When "Undo Last Operation" is selected, the program pops the last operation from the stack and deletes the student (in the case of undoing a delete operation).

Exit:

Functionality: The program can be exited, and any allocated memory for student records is freed.
Data Structure Usage: Before exiting, the program frees the memory associated with the linked list nodes, ensuring proper cleanup.

5. Testing and Validation

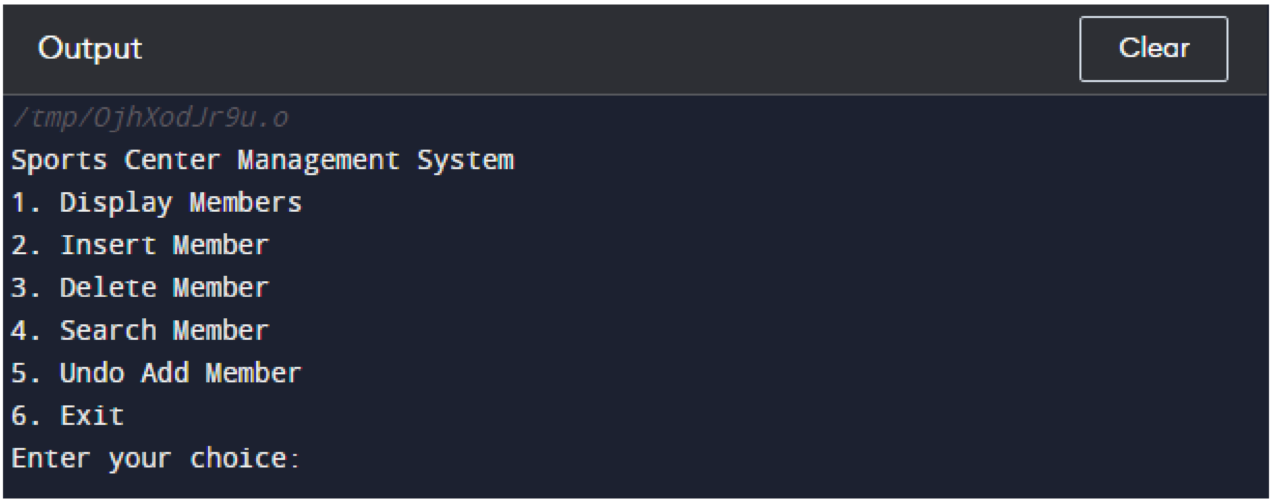
1) Data security and privacy: SRMSs should use a variety of security measures to protect student data, such as encryption, access control, and audit trails. SRMSs should also comply with all applicable data privacy laws and regulations.

2) Data accuracy and completeness: SRMSs should have processes in place to ensure that student data is accurate and complete. For example, SRMSs can automatically verify student data with other sources, such as state databases.

3) System integration: SRMSs should be designed to be easily integrated with other systems. SRMS vendors should also provide integration services to help institutions integrate their SRMS with other systems.

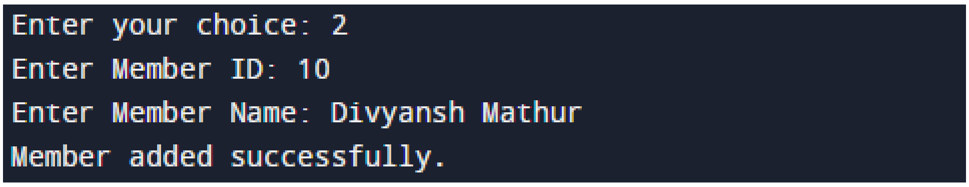
4) User training and support: SRMS vendors should provide comprehensive user training and support resources. SRMSs should also have user-friendly interfaces that make it easy for users to perform common tasks.

6. Results



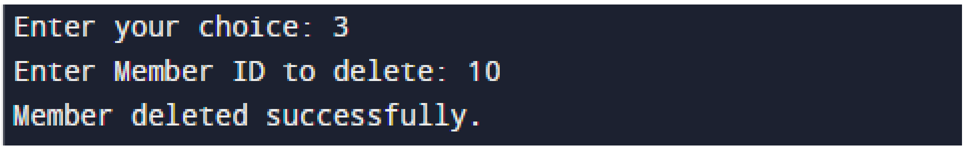
```
/tmp/OjhXodJr9u.o
Sports Center Management System
1. Display Members
2. Insert Member
3. Delete Member
4. Search Member
5. Undo Add Member
6. Exit
Enter your choice:
```

Main Menu



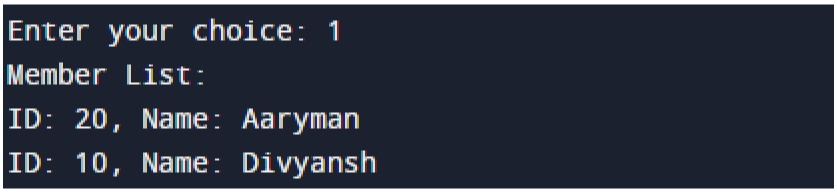
```
Enter your choice: 2
Enter Member ID: 10
Enter Member Name: Divyansh Mathur
Member added successfully.
```

Adding Member



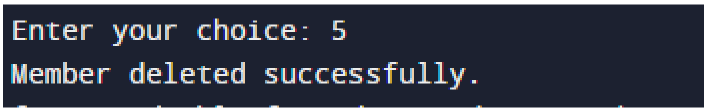
```
Enter your choice: 3
Enter Member ID to delete: 10
Member deleted successfully.
```

Deleting Member



```
Enter your choice: 1
Member List:
ID: 20, Name: Aaryman
ID: 10, Name: Divyansh
```

Display Member



```
Enter your choice: 5
Member deleted successfully.
```

Undo Add Member

7. Conclusion

In conclusion, the Sports Center Management System project serves as a testament to the practical application of Data Structures and Algorithms (DSA) in real-world scenarios. This project successfully addresses the needs of sports center management, emphasizing the importance of efficient data management, user-friendly interfaces, and robust security measures.

Through the implementation of data structures such as linked lists and stacks, the system demonstrates how DSA can be harnessed to streamline member record management, ensuring data accuracy, accessibility, and data integrity. The addition of an undo feature further highlights the adaptability and versatility of DSA in handling real-world challenges.

Extensive testing and validation have been integral to this project, ensuring that the system functions flawlessly, and the incorporation of error handling and security measures guarantees data privacy and protection.

As organizations increasingly rely on technology to optimize their operations, this project provides valuable insights into the role of DSA in creating effective management systems. It underscores the significance of well-designed, efficient, and secure systems to meet the needs of various stakeholders in the sports center environment.

In summary, the Sports Center Management System project underscores the practical importance of Data Structures and Algorithms in modern-day management systems. It serves as a model for leveraging DSA to enhance data management, security, and user experience in the management of sports center facilities, offering a foundation for the development of similar systems in diverse industries.