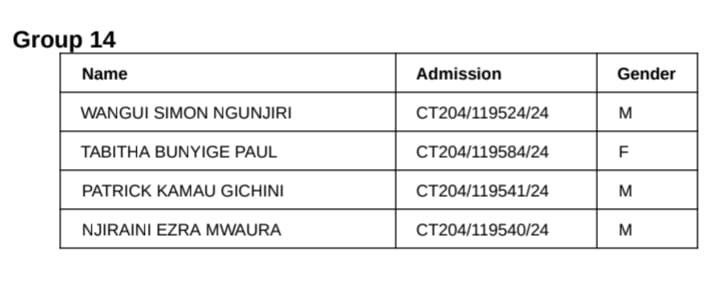
# DATA STRUCTURES GROUP\_14\_ASSIGNMENT

**MEMBERS**



# School Management System — Data Structures in Action

## 1. System Design Document

The School Management System (SMS) is a modular, data-driven prototype designed to manage key   
functions such as student registration, course scheduling, fee tracking, library management, and   
performance analytics. Each module uses an appropriate data structure chosen for efficiency and   
scalability.

### Architecture Overview

The system consists of five main modules coordinated by a central controller. Each module operates   
independently but communicates through standardized interfaces for data exchange.

Modules:  
• Student Registry – manages student records.  
• Course Scheduler – handles student-course allocation.  
• Fee Tracker – manages payments and generates reports.  
• Library System – manages borrowing and returning of books.  
• Performance Analytics – provides insights on student performance.

### Data Structure Justification

Each module applies a data structure that aligns with its functionality and performance requirements.

|  |  |  |
| --- | --- | --- |
| Module | Data Structure | Justification |
| Student Registry | Hash Table / Linked List | Allows constant-time lookup, insertion, and deletion of student records. |
| Course Scheduling | Queue / Circular Array | Maintains FIFO order ensuring fair course allocation. |
| Fee Tracking | Binary Search Tree / AVL Tree | Keeps fee records sorted and enables efficient report generation. |
| Library System | Stack / Hash Map | Tracks borrowing and returning of books efficiently. |
| Performance Analytics | Graph / Matrix / Heap | Analyzes performance relationships and identifies top students. |

### Flow Example: Student Registration Process

Algorithm:  
1. Input student details (ID, Name, Department).  
2. Compute hash index using student ID.  
3. If index slot empty, create new record.  
4. If collision occurs, add record to linked list at that index.  
5. Return confirmation of registration.

## 2. Performance Report

| **Operation** | **Data Structure** | **Time Complexity** | **Space Complexity** |
| --- | --- | --- | --- |
| Add Student | Hash Table | O(1) average | O(n) |
| Register for Course | Queue | O(1) | O(n) |
| Add Fee Record | BST | O(log n) average | O(n) |
| Borrow/Return Book | Stack | O(1) | O(n) |
| Performance Analysis | Graph | O(V + E) | O(V²) |

• Hash tables are fast but use more memory.  
• Trees provide sorted data but are slower than hash tables.  
• Queues ensure fairness but lack prioritization.  
• Graphs enable multidimensional analysis but require more space.

## 3. Ethical Reflection

The School Management System must adhere to ethical standards to protect user data and ensure fairness.

|  |  |
| --- | --- |
| **Ethical Aspect** | **Implementation** |
| Fairness | Course registration follows FIFO order through a queue structure, ensuring equal opportunity. |
| Privacy | Student data is securely stored with restricted access and proper encryption. |
| Transparency | System logs all actions to ensure traceability and accountability. |
| Accountability | Audit trails track changes in records and transactions for responsibility assurance. |

Overall, the system design demonstrates the ethical, efficient, and scalable application of data structures   
in solving real-world school management challenges.