CS3400 - Mini-Project: Data Structures in Action

Course: CS3400 Data Structures and Algorithms
Duration: 3 Weeks
Language: C++
Group Size: 4 Students

Project Overview

This mini-project aims to integrate the theoretical and practical aspects of data structures and algorithm design through the development of a real-world system. Each team will design, implement, test, and document a complete program demonstrating multiple data structures (arrays, linked lists, stacks, queues, trees, graphs, and hashing).

The project must show good design, logical planning, and implementation discipline, reflecting what you have learned throughout the course.

Learning Objectives

By the end of this project, students should be able to:
Apply various data structures appropriately to solve practical problems.
Perform algorithmic analysis (basic operation count, best/worst case).
Practice modular programming and system design.
Produce professional documentation and presentation.

Choose One of the Following Themes (or Propose Your Own)

1. NUL Campus Resource Management System

Manage booking and availability of university resources (rooms, labs, projectors).

Hash tables for login and user data.

Queues for booking/waitlists.

Trees for scheduling and searching.

Graph for campus map navigation.

2. NUL Smart Library System

Track book borrowing, reservations, and catalog searches.

BSTs/AVL Trees for indexing books.

Hashing for quick title/author lookup.

Queues for waitlisted titles.

Linked lists for transaction history.

3. Roma Mini Social Network

Simulate user profiles, connections, and interactions.

Graph for friendships.

Queues for feed updates.

Hash table for user accounts.

Linked list for message threads.

4. Maseru Transport & Navigation Planner

Plan routes and manage public transport operations.

Graph algorithms (BFS, DFS, Dijkstra).

Priority queues for route optimization.

Linked lists for route histories.

Hashing for station lookup.

5. St. Josephs Hospital Patient Management System

Handle patient registration, priority treatment, and bed assignment.

Priority queues for emergency cases.

Trees for room allocation.

Hash tables for patient data.

Linked lists for patient logs.

Each group may propose a different localised theme; subject to lecturer approval.

Project Timeline (3 Weeks)

Week Activities	**Deliverables**
1: Topic Selection, Requirements gathering, design and partial coding	ER Diagram, Architecture and Pseudo Code
2: Implementation, debugging and testing	Code
3: Presentation and Submission	Report and Presentation

Report format

Basic Details (Front Page)

- Project Title
- Course Name and Code
- Group Members (Names & IDs)
- Lecturer's Name
- Submission Date

1. Abstract (½ page)

A short summary (about 150 words) describing:

- What the system does
- Main data structures used
- Tools/language used
- Overall outcome

2. Introduction (1 page)

- Problem statement
- Project objectives
- Brief description of chosen theme
- Scope and limitations

3. System Design (2–3 pages)
Show how you planned your solution before coding. Include:

- Architecture diagram (blocks showing modules)
- Choice of data structures (table format is fine):

Feature	Data Structure	Justification
Booking list	Queue	FIFO order of requests
Search by ID	Hash Table	Fast access
Map routes	Graph	Represent connections

- Short pseudocode or flowcharts for 1–2 key operations

4. Implementation (2–3 pages)
Development environment and programming language
Description of key modules/classes
Sample screenshots or outputs
Explanation of one key algorithm (insert/search/delete)

5. Testing & Analysis (1–2 pages)
Short table of test cases (input, expected output, result)
Mention basic operation for one algorithm
Discuss best and worst-case complexity briefly

6. Results and Discussion (1 page)
Summary of what works and what could be improved
Lessons learned (technical and teamwork aspects)

7. Conclusion (½ page)
Main achievements
Possible future improvements

8. References (½ page)
Use a simple format (APA or IEEE). Example:
Knuth, D. (1997). The Art of Computer Programming. Addison-Wesley.

Formatting Guidelines

- Font: Times New Roman, 12pt
- Line spacing: 1.5
- Margins: Normal (2.54 cm)
- Number all pages
- Maximum length: 12 pages (excluding appendices)