Data Structures and Algorithms

HEALTHCARE MANAGEMENT SYSTEM

Course Project Report

School of Computer Science and Engineering

2024-25

Contents

1. Course and Team Details

1.1 Course details

Course Name	Design and Analysis of Algorithms
Course Code	24ECSC205
Semester	III
Division	A
Year	2024-25
Instructor	Mallikarjun Akki

1.2 Team Details

Si. No.	Roll No.	Name
1.	118	Prasad Morab
2.	120	Prasanna Anagal
3.	137	Siddharth Prabhu
4.	140	Shashank Joshi

1.3 Report Owner

Roll No.	Name
118	Prasad Morab

2. Introduction

The Healthcare Management System (HMS) is a project that uses efficient algorithms and data structures to optimize healthcare operations. In a city, health of citizens matters the most. So this project is picked although there were many problems that may be addressed. HMS addresses four key areas:

Emergency Incident Categorization – categorizing and prioritizing medical emergencies;
Patient Appointment Scheduling – scheduling appointments for patients in an efficient manner;
Medical Supply Inventory Management – tracking and restocking of medical supplies;
Healthcare Resource Allocation – optimizing the critical distribution of healthcare resources.
HMS is done with the help of algorithms of Dijkstra's Algorithm, the Rabin-Karp string matching, QuickSort, Depth First Search, and Priority Queues, BST, and Union Find to bring improved efficiency in timely real-time decision making.

It is based on the principles from algorithmic problem-solving and data structures, discussed in key references like Data Structures and Algorithm Analysis in C++ by Weiss [1] and Introduction to Algorithms by Cormen et al. [2]. This helped choose suitable algorithms for every module, scalable enough for the healthcare scenario, while being very efficient in practice.

3. Problem Statement

3.1 Domain

The HMS automates healthcare operations such as incident management, patient scheduling, resource allocation and inventory management. These tasks are full of inefficiencies and delays, leading to poor service quality and increased costs. The system's objective is to resolve these challenges by providing an algorithm-driven, scalable, and user-friendly solution.

3.2 Module Description

The **Medical Supply Inventory Management System** ensures efficient tracking and restocking of essential medical supplies. This system optimizes inventory management using **hashing, tree-based search, and sorting techniques** to ensure smooth operations in hospitals and healthcare centers.

Key Algorithms Used:

1. **Hashing** – Enables quick lookup of inventory items.
2. **Binary Search Tree (BST)** – Organizes inventory for efficient searches and updates.
3. **Selection Sort** – Prioritizes items for restocking based on demand and urgency.

4. Functionality Selection

Si. No.	Functionality Name	Known	Unknown	Principles applicable	Algorithms	Data Structures
	Name the functionality within the module	What information do you already know about the module? What kind of data you already have? How much of process information is known?	What are the pain points? What information needs to be explored and understood? What are challenges?	What are the supporting principles and design techniques?	List all the algorithms you will use	What are the supporting data structures?
1	Inventory Lookup	Each medical supply item has a unique identifier. Lookup needs to be quick.	Need an efficient way to store and retrieve inventory items in constant time.	Hashing Principles	Hashing	Hash Tables
2	Inventory Organization	Supplies should be stored in a structured manner for efficient searches and updates.	Large datasets require a structured format to avoid slow lookup times.	Binary Search Trees	Binary Search Tree (BST)	Trees
3	Restocking Prioritization	Some supplies are more critical than others and should be restocked first.	Sorting should be efficient and prioritize high-usage items.	Selection Approach	Selection Sort	Arrays

5. Functionality Analysis

5.1 Inventory Lookup (Hashing)

- **Workflow:**
  1. Each inventory item is assigned a unique key using a hash function.
  2. Items are stored in a hash table for fast access.
  3. When checking stock, the system retrieves the item in O(1) average time.
- **Efficiency:**
  - **Time Complexity:** O(1) (average case), O(n) (worst case with collisions).
  - **Space Complexity:** O(n) for storing items in the hash table.

5.2 Inventory Organization (Binary Search Tree - BST)

- **Workflow:**
  1. Inventory items are stored in a BST, sorted by name or ID.
  2. Search, insert, and delete operations are performed efficiently.
  3. The tree remains balanced for optimal performance.
- **Efficiency:**
  - **Time Complexity:** O(log n) (balanced BST), O(n) (worst case).
  - **Space Complexity:** O(n).

5.3 Restocking Prioritization (Selection Sort)

- **Workflow:**
    1. Items are sorted based on urgency and demand using Selection Sort.
    2. The highest-priority items are identified for restocking.
    3. The sorted order ensures critical items are replenished first.
- **Efficiency:**
    - **Time Complexity:** O(n^2).
    - **Space Complexity:** O(1).

6. Conclusion

The Medical Supply Inventory Management System enhances hospital inventory efficiency by ensuring that essential medical supplies are tracked, retrieved, and restocked efficiently. The integration of Hashing for quick lookup, Binary Search Trees (BST) for structured storage, and Selection Sort for prioritizing restocking creates a robust system for preventing shortages and optimizing supply chain management.
This module highlights the importance of efficient data structures in real-world inventory management, where fast access and sorting of critical items are vital. By applying sorting techniques for restocking and structured searching for retrieval, this system ensures seamless inventory control in hospitals and medical facilities.

7. References

[1] **Weiss, Mark Allen.** *Data Structures and Algorithm Analysis in C++ (4th Edition).* Pearson, 2013.
[2] **Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson.** *Introduction to Algorithms (3rd ed.).* The MIT Press, 2009.

~*~*~*~*~*~*~*~