

Clinic Inventory System

Final Project Proposal

Course Code: CPE 201L	Program: BSCPE
Course Title: Data Structures and Algorithms	Date Performed:
Section: CpE-2B	Date Submitted:
Leader: Aquino, Jester J.	
Members: Caasi, Karl Elpedes, Glen Monoy, Justin Rhey Tan, Charles	Instructor: Engr. Maria Rizette Sayo
1. Objective(s):	
The main goal of this project is to design and develop an efficient system that accurately manages and monitors the clinic's medical supplies, equipment, and resources.	
This is accomplished by two major objectives:	
<ul style="list-style-type: none">• To efficiently manage and organize clinic inventory to ensure fast storage, retrieval, and updating of medical supply records.• To automate and optimize the inventory process minimizing human errors, improving accuracy, and providing real-time information for better decision-making and resource management.	
2. Intended Learning Outcomes (ILOs):	
<ul style="list-style-type: none">• Understand the concept and importance of using arrays in managing and organizing clinic inventory data such as medicines, equipment, and supplies.• Apply basic data structure principles, particularly arrays, to store, retrieve, and update inventory records efficiently.• Adding a simple script to handle inventory transactions like adding, deleting, and searching.• Strengthen skills in Python programming and system design through the practical use of arrays.• Analyze the efficiency and accuracy of the system in managing real-world clinic operations.	

3. Discussion:

The Clinic Inventory System is designed to help our medical facilities manage their supplies, medicines, and equipment efficiently. In many clinics, manual inventory tracking often leads to errors, misplaced items, and inaccurate stock levels. Through the use of a computerized system, managing inventory becomes more organized, accurate, and time-saving.

This system allows users to record essential information such as item names, quantities, expiration dates, and categories. It also provides basic functions like adding new items, updating existing records, searching for specific items, and deleting outdated entries. By automating these processes, the system minimizes human error and improves record accuracy.

Moreover, this project enhances students' understanding of how data can be structured, stored, and managed within a system. It demonstrates how programming and logical organization can be applied to solve real-world problems in the healthcare setting. The Clinic Inventory System not only promotes efficiency in operations but also ensures that clinics maintain adequate supplies to provide continuous and effective patient care.

4. Materials and Equipments:

- Computer
- Python
- Cursor
- Github

5. Procedure:

Step 1: Start the application (cursor).

Step 2: Get the tools ready by importing all tools needed to build the program

Step 3: Make it look nice by sets the color theme and appearance

Step 4: Create an empty storage boxes like two empty list to store data (example medicine and equipment)

Step 5: Add a New medicine like adding new product to your inventory list

Step 6: Insert Medicine at the Beginning

Instead of adding at the end, you can add at position 0 (the very beginning)

all other medicines get pushed down

Step 7: Remove Medicine by ID

Step 8: Update Existing Medicine

Program loads the medicines data into the input fields

Step 9: Search by name

Step 10: Find by ID

Program goes through the list one by one

Check if ID matches

Returns the medicine when found

Step 11: Sort Medicines

Programs rearranges the medicine list alphabetically

Table refreshes to show sorted order

Step 12: Create Tabs

Create two tabs Medicine and Equipment

Step 13: Adding Input fields

Step 14: Highlighting Low Stock

Program checks each medicine in the table

If packs less than or total_qty less than 5 makes row Red

If expiring within 30 days make row yellow

Step 16: Activity Log

Every action is logged with a timestamp

Step 17: Real time clock

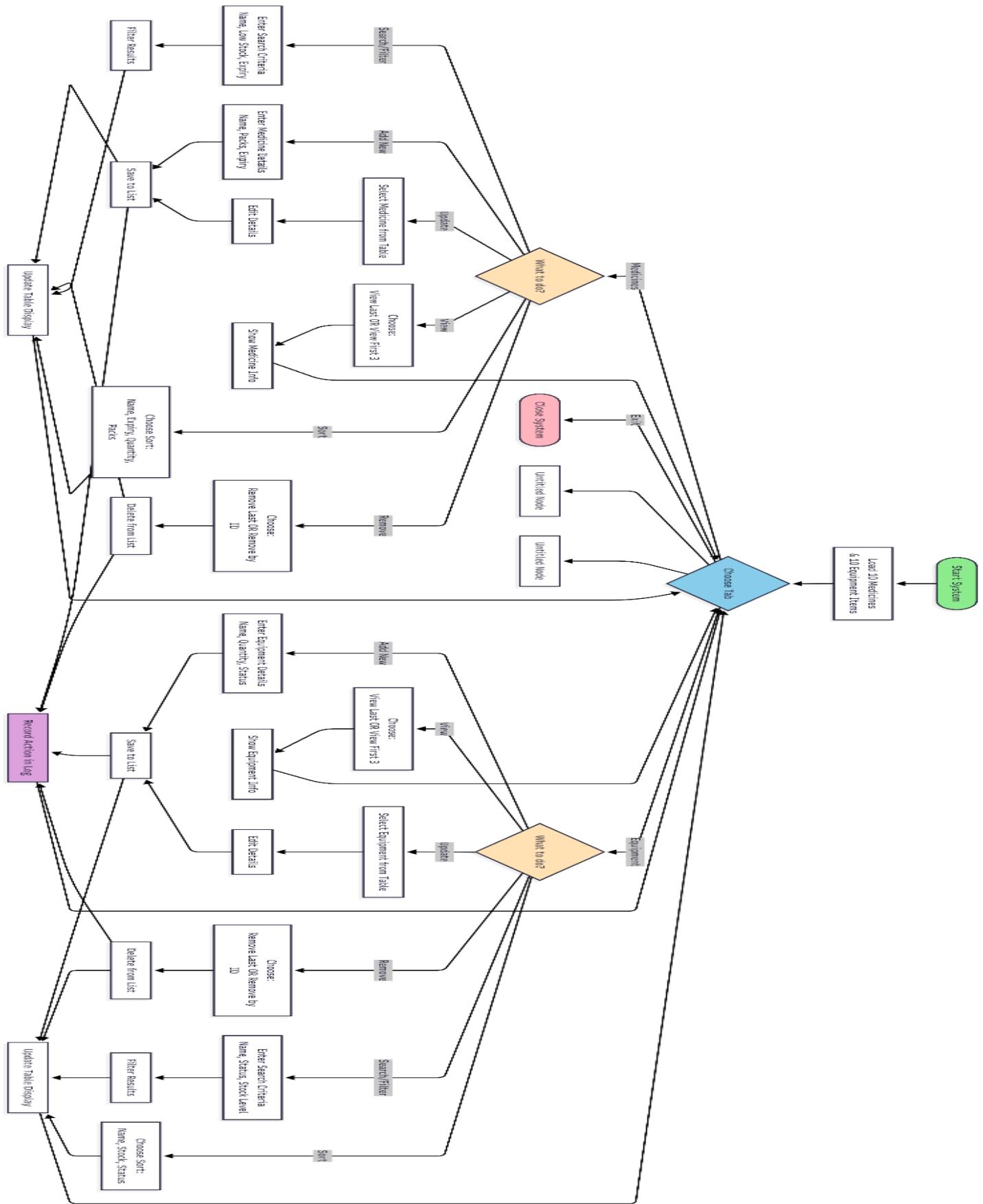
Clock in top right corner updates every second

Step 14: End (If no refresh is triggered, the app becomes idle and waits for user input, which restarts the loop at Step 9).

6. Algorithm

1. Start the program
2. Initialize Medicine[] and Equipment[] arrays
3. If “Medicines” is chosen
4. Display Options
 - Add New
 - Update
 - Remove
 - Search/Filter
 - Sort
 - View
5. If Add:
 - Input item details
 - Add item to array
6. If Update:
 - Input item name
 - Search in array
- Update details if found
7. If Delete:
 - Input item name
 - Search in array
- Remove item if found
8. If Search:
 - Input item name
 - Search in array
- Display details if found
9. If Display:
 - Show all items in arrays
10. Reset to exit
11. End

7. FlowChart



8. Pseudocode

BEGIN

 Initialize system

 Input tabChoice // Medicines, Equipment, or Exit

 IF tabChoice = Medicines THEN

 Input medChoice // Add, Update, Remove, Search/Filter, Sort, View, or Back

 IF medChoice = Add THEN

 Input name, packs, expiry

 Add (name, packs, expiry) to MedicineList

 UpdateTableDisplay()

 ELSE IF medChoice = Update THEN

 Input selectedMedicine

 Input newName, newPacks, newExpiry

 Update selectedMedicine with (newName, newPacks, newExpiry)

 UpdateTableDisplay()

 ELSE IF medChoice = Remove THEN

 Input removeChoice // Remove by ID or Remove Last

 IF removeChoice = Remove by ID THEN

 Input medicineID

 Delete medicine with ID = medicineID from MedicineList

 ELSE

 Remove last item from MedicineList

 ENDIF

 UpdateTableDisplay()

 ELSE IF medChoice = Search/Filter THEN

 Input criteria // Name, Low Stock, Expiry

 Filter MedicineList using criteria

 UpdateTableDisplay()

 ELSE IF medChoice = Sort THEN

 Input sortChoice // Name, Expiry, Quantity, Packs

 Sort MedicineList by sortChoice

 UpdateTableDisplay()

 ELSE IF medChoice = View THEN

 Input viewChoice // View Last or View First 3

 IF viewChoice = View Last THEN

 Show last Medicine item(s)

```
ELSE
    Show first 3 Medicine items
ENDIF
ENDIF
```

```
ELSE IF tabChoice = Equipment THEN
    Input eqChoice // Add, Update, Remove, Search/Filter, Sort, View, or Back
```

```
IF eqChoice = Add THEN
    Input name, quantity, status
    Add (name, quantity, status) to EquipmentList
    UpdateTableDisplay()
    RecordActionInLog()
```

```
ELSE IF eqChoice = Update THEN
    Input selectedEquipment
    Input newName, newQuantity, newStatus
    Update selectedEquipment with (newName, newQuantity, newStatus)
    UpdateTableDisplay()
    RecordActionInLog()
```

```
ELSE IF eqChoice = Remove THEN
    Input removeChoice // Remove by ID or Remove Last
    IF removeChoice = Remove by ID THEN
        Input equipmentID
        Delete equipment with ID = equipmentID from EquipmentList
    ELSE
        Remove last item from EquipmentList
    ENDIF
    UpdateTableDisplay()
    RecordActionInLog()
```

```
ELSE IF eqChoice = Search/Filter THEN
    Input criteria // Name, Status, Stock Level
    Filter EquipmentList using criteria
    UpdateTableDisplay()
    RecordActionInLog()
```

```
ELSE IF eqChoice = Sort THEN
    Input sortChoice // Name, Stock, Status
    Sort EquipmentList by sortChoice
    UpdateTableDisplay()
    RecordActionInLog()
```

```

ELSE IF eqChoice = View THEN
    Input viewChoice // View Last or View First 3
    IF viewChoice = View Last THEN
        Show last Equipment item(s)
    ELSE
        Show first 3 Equipment items
    ENDIF
ENDIF

```

```

ELSE IF tabChoice = Exit THEN
    Stop system
ENDIF

```

END

9. Output

The screenshot shows a Windows application window titled "Clinic Inventory System". At the top, there are tabs for "Medicines" and "Equipment", with "Equipment" being the active tab. The status bar indicates the time as "Time: 17:39:10". Below the tabs, there are input fields for "Name", "Quantity", and "Description / Location (optional)". Buttons for "Add Equipment", "Insert First", "Update Equipment", "View Last", "View First 3", and "Remove Last" are visible. There are also buttons for "Remove by ID", "Remove by ID", and "Remove Last". Below these are dropdowns for "Sort by" and "Filter", and a search bar with a "Search" button. The main area displays a table of equipment data:

Id	Name	Quantity	Description
1	Stethoscope	8	Available
2	Digital Thermometer	12	Available
3	Blood Pressure Monitor	6	Available
4	Oxygen Tank	0	Out of Stock
5	Nebulizer Machine	3	In Use
6	ECG Machine	2	Available
7	Defibrillator	1	Available
8	Surgical Gloves (Box)	25	Available
9	Syringes (Box)	30	Available
10	First Aid Kit	5	Available

At the bottom, a log message reads: "Recent Activity Log: [17:37:21] Application started."

Clinic Inventory System

Time: 17:38:32

Medicines Equipment

Name	Packs	Items / Pack	Total Quantity	Expiry (YYYY-MM-DD)		
Medicine name	e.g. 5	e.g. 10		YYYY-MM-DD		
+ Add Medicine		+ Insert First	Update Medicine	View Last	View First 3	
Remove by ID: ID		Remove by ID	Remove Last			
Sort by:	name	asc	Sort	Filter: name	Filter	Reset
<input type="text" value="Search medicines by name"/> Search Reset						
Id	Name	Packs	Items_per_pack	Total_qty	Expiry	
1	Paracetamol 500mg	20	10	200	2026-12-31	
2	Ibuprofen 400mg	15	12	180	2025-11-10	
3	Aspirin 100mg	25	8	200	2027-01-15	
4	Amoxicillin 500mg	10	14	140	2026-10-20	
5	Cetirizine 10mg	18	10	180	2027-02-28	
6	omeprazole 20mg	12	7	84	2026-09-30	
7	Metformin 500mg	22	10	220	2025-11-5	
8	Atorvastatin 20mg	8	14	112	2026-12-20	
9	Salbutamol Inhaler	5	1	5	2025-11-20	
10	Diclofenac 50mg	16	10	160	2026-11-15	

Recent Activity Log: [17:37:21] Application started.

7. Conclusion:

The Clinic Inventory System provides an organized and efficient way to manage the supplies, medicines, and equipment used in a clinic. It ensures that all inventory records are properly stored, updated, and accessed when needed, helping maintain accuracy and reliability in daily operations.

This project also highlights the importance of technology in modern healthcare operations. By automating inventory tasks, clinics can save time, minimize human error, and focus more on providing quality patient care. Overall, the Clinic Inventory System demonstrates how computer-based solutions can greatly enhance the daily operations of a medical facility.

8. References

- [1] GeeksforGeeks. (2025, September 10). *Getting Started with Array Data Structure*.
<https://www.geeksforgeeks.org/introduction-to-arrays-data-structure-and-algorithm-tutorials/>
- [2] W3Schools. (n.d.). *DSA Arrays*.
https://www.w3schools.com/dsa/dsa_data_arrays.php
- [3] Simplilearn. (2024, [n.d.]). *What Is Array in Data Structure? Types & Syntax*.
<https://www.simplilearn.com/tutorials/data-structure-tutorial/arrays-in-data-structure>
- [4] DEV Community. (2023, [n.d.]). *Array Data Structure*
<https://dev.to/ggorantala/array-data-structure-4cl2>
- [5] Masai School Blog. (2021, ?). *Array Data Structure – Types, Applications & Implementation*.
<https://www.masaischool.com/blog/array-data-structure-explained-with-examples/>