UPES
UNIVERSITY OF TOMORROW

C Programming
Major Project

NAME	**AKSHIT SINHA**
SAP ID	**590025645**
BATCH	**38**
COURSE CODE	**CSEG1032**
INSTRUCTOR	**Dr. Tanu Singh**

ABSTRACT

The C programming language was used to create the straightforward console-based Lost and Found System. The project's goal is to give users a systematic way to keep track of things they find in an institution and assist them in looking for things they might have misplaced. Users can withdraw an item after it has been claimed, report found items, search for lost ones by entering the exact name, and view all stored entries.

With no use of sophisticated library functions, this project's design is beginner-friendly, concentrating only on fundamental ideas like arrays, structures, loops, conditionals, and basic string comparison. This project aims to build a working system that mimics a real-world application while strengthening fundamental knowledge of C. All things considered, the project shows how basic programming concepts can be applied to solve common problems in an easy-to-use yet efficient way.

PROBLEM DEFINITION

Lack of a systematic procedure causes lost items to be improperly stored or retrieved, which is a persistent issue for many institutions. Students may have trouble confirming whether their misplaced possessions have been located, or items may get lost.

The project's goal is to create a straightforward Lost and Found management system that:

1) keeps track of the details of items found.

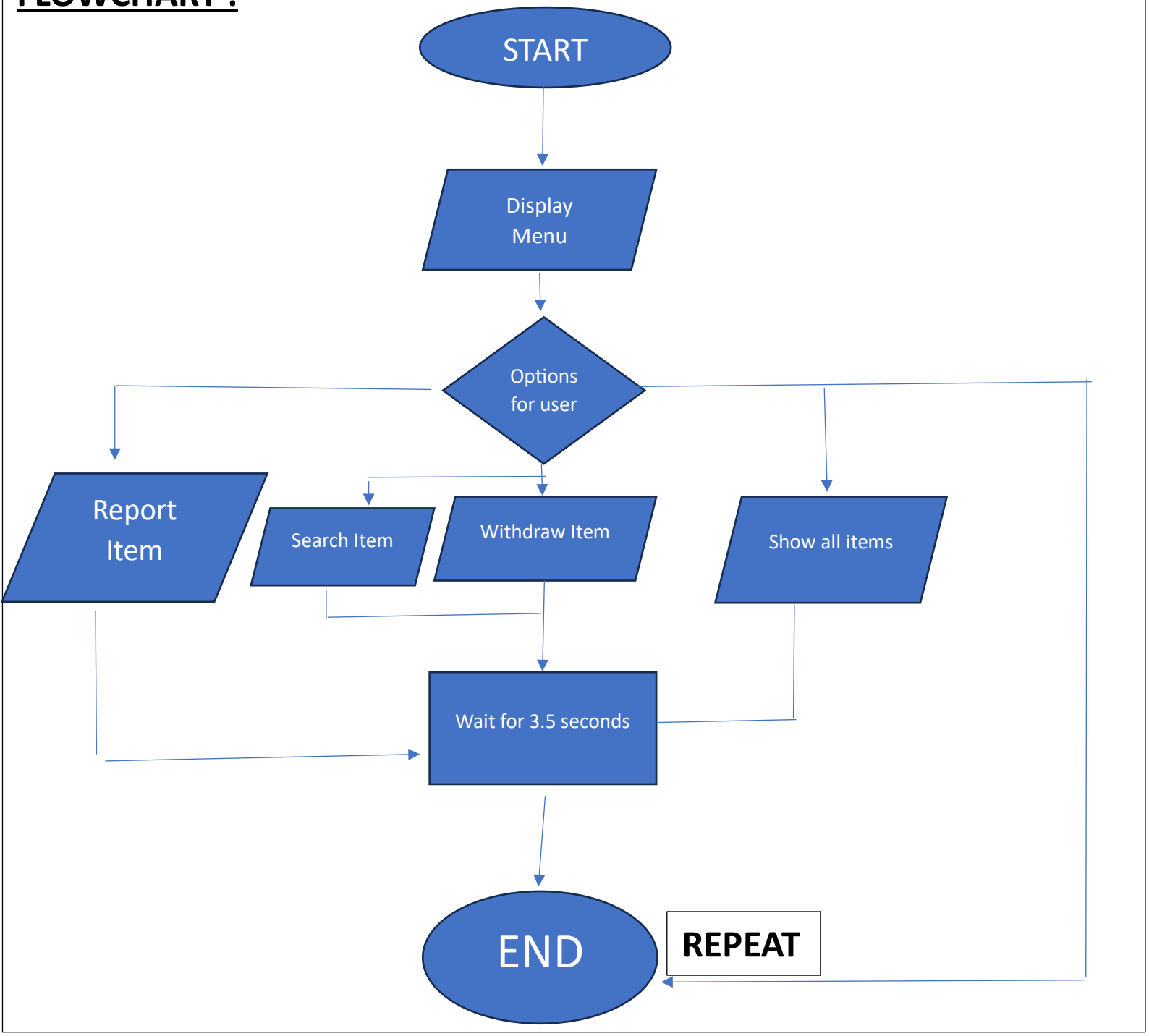
2) enables users to look for misplaced items

3) removes items from the system to aid in their retrieval.

4) enables the listing of every item that is currently stored.
The system must be simple to use, accessible to novices, and incorporate features utilizing only fundamental C programming principles.

SYSTEM DESIGN

FLOWCHART :



ALGORITHM

Step 1: Start the program and initialize an array of items where each item contains: ID, name, color, type, and presence flag.
Step 2: Display the main menu with five options: Report, Search, Withdraw, Show All, and Exit.
Step 3: Ask the user to enter their choice.
Step 4: If the user selects:

- **Report Item:**
  - Find an empty slot
  - Take inputs for name, color, type
  - Assign a new ID and mark the item as present
- **Search Item:**
  - Take name as input
  - Compare with stored names using manual character matching
  - Display matching records
- **Withdraw Item:**
  - Ask for ID
  - If the ID exists, mark the item as removed
- **Show All Items:**
  - Display all stored items
- **Exit:**
  - End the program

Step 5: After completing any action, wait for 3.5 seconds.
Step 6: Return to the main menu.
Step 7: Continue until the user selects the Exit option.

4. IMPLEMENTATION DETAILS

- **Structures (struct) to group item details**
- **Arrays to store multiple items**
- **Loops for menu repetition and item searching**
- **If–else conditions for menu operations**
- **A manually written string comparison function**
- **Delay function (Sleep(3500))**
- **Basic C I/O (printf, scanf)**

Code Snippet

```
int same(char a[], char b[]) {
    int i = 0;
    while (1) {
        if (a[i] != b[i]) return 0;
        if (a[i] == '\0') return 1;
        i++;
    }
}
```

TESTING & RESULTS

The system was tested with several sample inputs to ensure that all functional requirements were successfully met.

Test Case 1 — Adding a Found Item

Input:
Name: wallet
Colour: black

Type: leather
Expected Output:
“Item saved with ID 1”
Result: PASS

Test Case 2 — Searching an Existing Item

Input:
Name: wallet
Expected Output:
Item details displayed
Result: PASS

Test Case 3 — Searching an Invalid Item

Input:
Name: charger
Expected Output:
“No item found.”
Result: PASS

Test Case 4 — Withdrawing an Item

Input:
ID: 1

Expected Output:
“Item withdrawn.”
Result: PASS

Test Case 5 — Listing All Items

Expected:
Show remaining items
Result: PASS

6. CONCLUSION & FUTURE WORK

The Lost and Found System is a good way to learn basic C programming concepts through a small project that works. The system fixes the main problem of organizing lost items so that they can be found and stored.

Future Enhancements :

1) Adding file handling to save data permanently
2) Allowing multi-word item names

3) Improving search options (search by color, type, or ID)

4) Adding a graphical interface

5) Allowing admin login or password protection

This project provides a good foundation and can easily be expanded into a more advanced management system.

REFERENCES

1) Let us C book

2) Online resources

3) Class notes & PPT’s

4) C documentations