



UNIVERSITAS
INDONESIA
Veritas, Praelatio, Ineditio

FAKULTAS

TEKNIK

Background

Objectives

Limitation

User

Flowchart & Code

PharmaSys

One System For All Your Pharmaceutical Needs

by Group 3



Raisa Shahira A.
(2306168971)



Reyhana Arini P.
(2306168990)



Lukas Sudibyo P.
(2306264603)

Praktikum Pemrograman C

Background

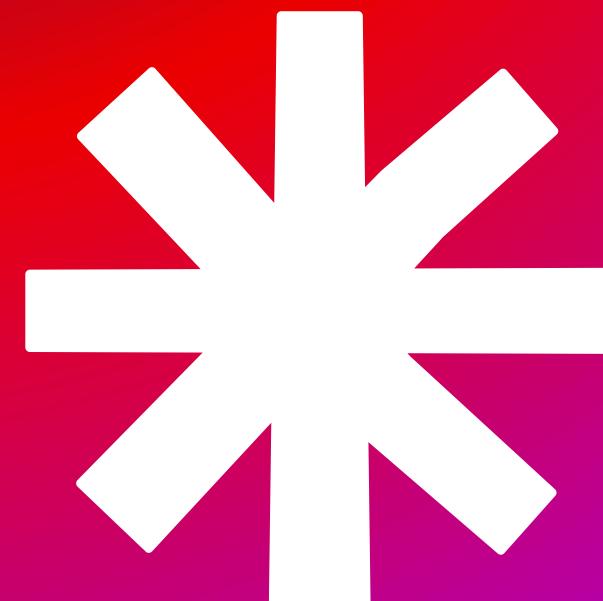
Mengapa perlu **PharmaSys**?

- Sistem manajemen obat manual menyulitkan pengelolaan data dan pencarian informasi
- Keamanan data yang rendah karena tidak ada kontrol akses
- Risiko terjadi human error



One system for all
your pharmaceutical
needs

One system for
all
your
pharmaceutical
needs



Objectives

- Meningkatkan efisiensi pengelolaan obat.
- Menjaga akses dan keamanan melalui autentikasi.
- Mempermudah pencatatan dan pelaporan semua data karena tersimpan dalam file lokal.
- Penggunaan sistem modular untuk kemudahan perawatan dan pengembangan program.

Limitations

✓ Low Password Security

Password disimpan dalam file teks biner (users.txt) tanpa enkripsi.

✓ No Database System

Data hanya disimpan pada file lokal (.txt), yang membuatnya tidak cocok untuk sistem skala besar.

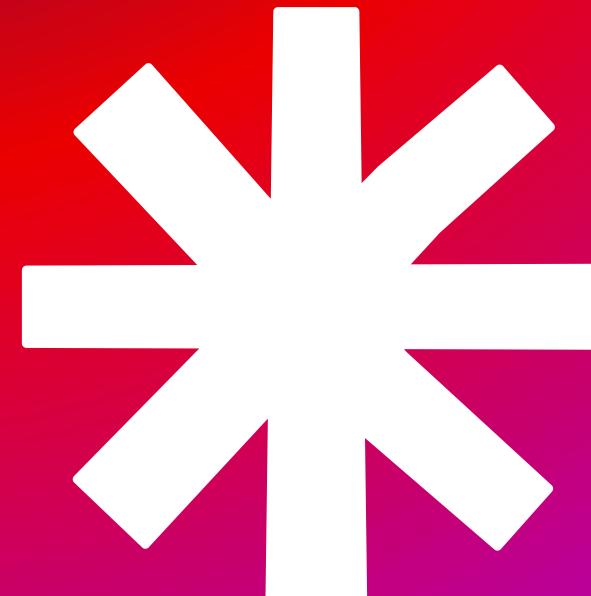
✓ Limited Validation

Tidak ada pengecekan format username, atau handling kesalahan input yang robust.

✓ Does not support multi-user

Tidak ada real-time update, lalu jika dua user mengubah database, akan menyebabkan file corruption

One system for
all
your
pharmaceutical
needs



User

- Pemilik/Manager Apotek
- Kasir Apotek

Pseudocode

TAMPILKAN menu login
PANGGIL loginMenu()

JIKA login berhasil SEBAGAI manager
TAMPILKAN menu manager

PILIHAN:

- Tambah obat
- Lihat daftar obat
- Update obat
- Hapus obat
- Logout

JIKA login berhasil SEBAGAI kasir

TAMPILKAN menu kasir

PILIHAN:

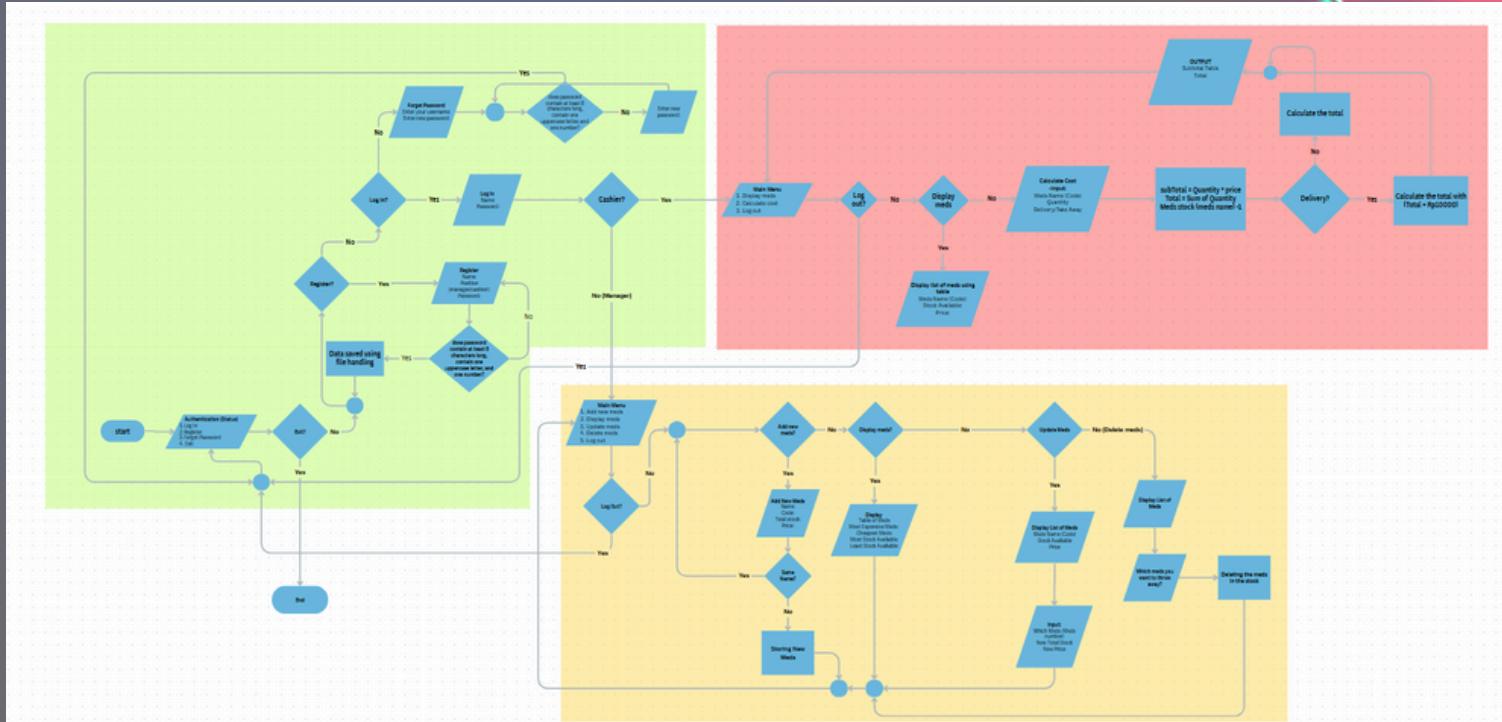
- Lihat daftar obat
- Lakukan transaksi pembelian
- Logout

JIKA user memilih Exit

KELUAR dari program

END PROGRAM

Main File



```

#include <stdio.h>
#include <windows.h>
#include <stdlib.h>
#include <string.h>
#include "auth.h" //masukin modul2 yang terpisah
#include "meds.h"
#include "utils.h"
#include "menu.h"

void showWelcomeScreen() {
    printf("=====\\n");
    printf("   Welcome to PharmaSys!   \\n");
    printf("=====\\n");
    printf(" Your trusted companion in managing and\\n");
    printf(" purchasing medications with ease and safety.\\n\\n");
    printf(" Stay healthy. Stay in control.  \\n\\n");
    printf(" Press Enter to continue...\\n");
    getchar(); //sebenarnya buat masukin karakter sebelum user klik enter, tapi ini buat cuma buat user klick enter
}

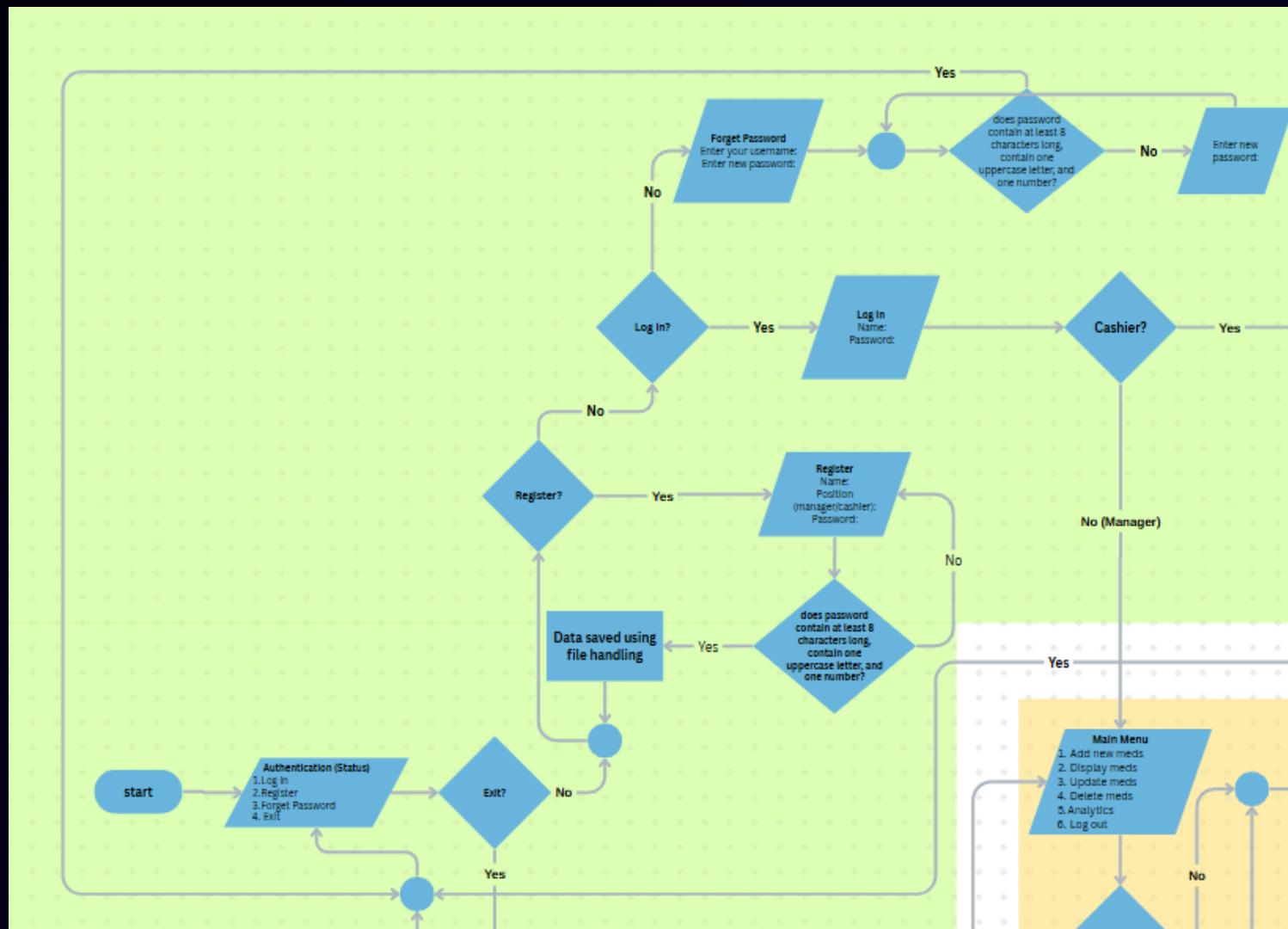
int main() {
    SetConsoleOutputCP(65001); //ini supaya emoji bisa dilihat di terminal menggunakan skema pengkodean karakter(UTF 8)
    showWelcomeScreen();
    int status;

    while (1) {
        status = loginMenu(); //ngambil nilai status yang direturn dari fungsi menu
        if (status == -1) break; //kalo dia case 3 programnya di brake
        else if (status == 1) cashierMenu(); //login sebagai apa?
        else if (status == 2) managerMenu();
    }
    return 0;
}

```

*User Authorization

Auth.h



```

#ifndef AUTH_H
#define AUTH_H
#include "utils.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

typedef struct {
    char username[50];
    char password[50];
    int isManager;
} User;

int loginMenu();
  
```

```

int isValidPassword(const char *password) {
    int len = strlen(password);
    if (len < 8) return 0;

    int hasUpper = 0, hasDigit = 0;
    for (int i = 0; i < len; i++) {
        if (isupper(password[i])) hasUpper = 1;
        if (isdigit(password[i])) hasDigit = 1;
    }

    return hasUpper && hasDigit;
}
  
```

Function login()

```

int login() {
    char username[50], password[50];
    int attempts = 0;

    while (attempts < 3) {
        FILE *file = fopen("users.txt", "rb");
        if (!file) return 0;
        User u;

        printf("Username: "); fgets(username, sizeof(username), stdin);
        printf("Password: "); fgets(password, sizeof(password), stdin);

        username[strcspn(username, "\n")] = '\0';
        password[strcspn(password, "\n")] = '\0';

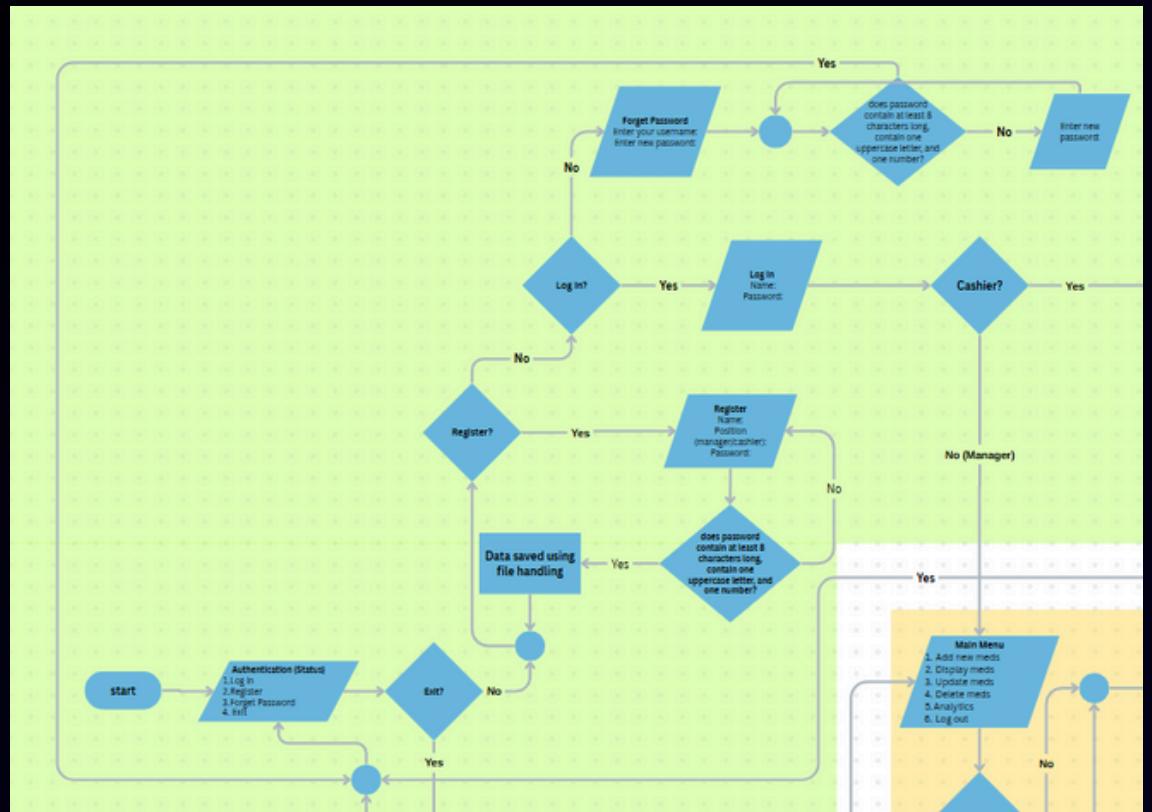
        while (fread(&u, sizeof(User), 1, file)) {
            if (strcmp(u.username, username) == 0 && strcmp(u.password, password) == 0) {
                fclose(file);
                return u.isManager ? 2 : 1;
            }
        }
        fclose(file);
        printf("Login failed. Try again.\n");
        attempts++;
    }

    printf("Too many failed attempts.\nDo you want to reset your password? (y/n): ");
    char choice;
    scanf(" %c", &choice); getchar();
    if (choice == 'y' || choice == 'Y') {
        resetPassword(username);
    }
    printf("Returning to menu. Press Enter..."); getchar(); clearScreen();
    return loginMenu();
}
  
```

Function resetPassword()

```

int resetPassword(const char *username) {
    FILE *file = fopen("users.txt", "rb+");
    if (!file) return 0;
    User u;
    long pos;
    while ((pos = ftell(file)), fread(&u, sizeof(User), 1, file)) {
        if (strcmp(u.username, username) == 0) {
            char newPassword[50];
            do {
                printf("Enter new password: ");
                fgets(newPassword, sizeof(newPassword), stdin);
                newPassword[strcspn(newPassword, "\n")] = '\0';
                if (!isValidPassword(newPassword)) {
                    printf("Password must be at least 8 characters long, contain one uppercase letter, and one digit.\n");
                }
            } while (!isValidPassword(newPassword));
            strcpy(u.password, newPassword);
            fseek(file, pos, SEEK_SET);
            fwrite(&u, sizeof(User), 1, file);
            fclose(file);
            printf("Password reset successful!\n");
            return 1;
        }
    }
    fclose(file);
    printf("Username not found.\n");
    return 0;
}
  
```



Function login()

```

int login() {
    char username[50], password[50];
    int attempts = 0;

    while (attempts < 3) {
        FILE *file = fopen("users.txt", "rb");
        if (!file) return 0;
        User u;

        printf("Username: "); fgets(username, sizeof(username), stdin);
        printf("Password: "); fgets(password, sizeof(password), stdin);

        username[strcspn(username, "\n")] = '\0';
        password[strcspn(password, "\n")] = '\0';

        while (fread(&u, sizeof(User), 1, file)) {
            if (strcmp(u.username, username) == 0 && strcmp(u.password, password) == 0) {
                fclose(file);
                return u.isManager ? 2 : 1;
            }
        }
        fclose(file);
        printf("Login failed. Try again.\n");
        attempts++;
    }

    printf("Too many failed attempts.\nDo you want to reset your password? (y/n): ");
    char choice;
    scanf("%c", &choice); getchar();
    if (choice == 'y' || choice == 'Y') {
        resetPassword(username);
    }
    printf("Returning to menu. Press Enter..."); getchar(); clearScreen();
    return login();
}
  
```

Function loginMenu()

```

int loginMenu() {
    int choice;
    char username[50];
    do {
        printf("==== AUTHENTICATION ====\n1. Login\n2. Register\n3. Forgot Password\n4. Exit\nChoice: ");
        scanf("%d", &choice); getchar(); clearScreen();
        switch (choice) {
            case 1: return login();
            case 2: registerUser(); break;
            case 3:
                printf("Enter your username: ");
                fgets(username, sizeof(username), stdin);
                username[strcspn(username, "\n")] = '\0';
                resetPassword(username);
                break;
            case 4: return -1;
        }
    } while (1);
}

#endif
  
```

Function registerMenu()

```

void registerUser() {
    User u;
    char tempPassword[50];
    printf("New username: "); fgets(u.username, sizeof(u.username), stdin);
    u.username[strcspn(u.username, "\n")] = '\0';

    do {
        printf("New password: "); fgets(tempPassword, sizeof(tempPassword), stdin);
        tempPassword[strcspn(tempPassword, "\n")] = '\0';
        if (!isValidPassword(tempPassword)) {
            printf("Password must be at least 8 characters long, contain one uppercase letter, and one number.\n");
        }
    } while (!isValidPassword(tempPassword));

    strcpy(u.password, tempPassword);

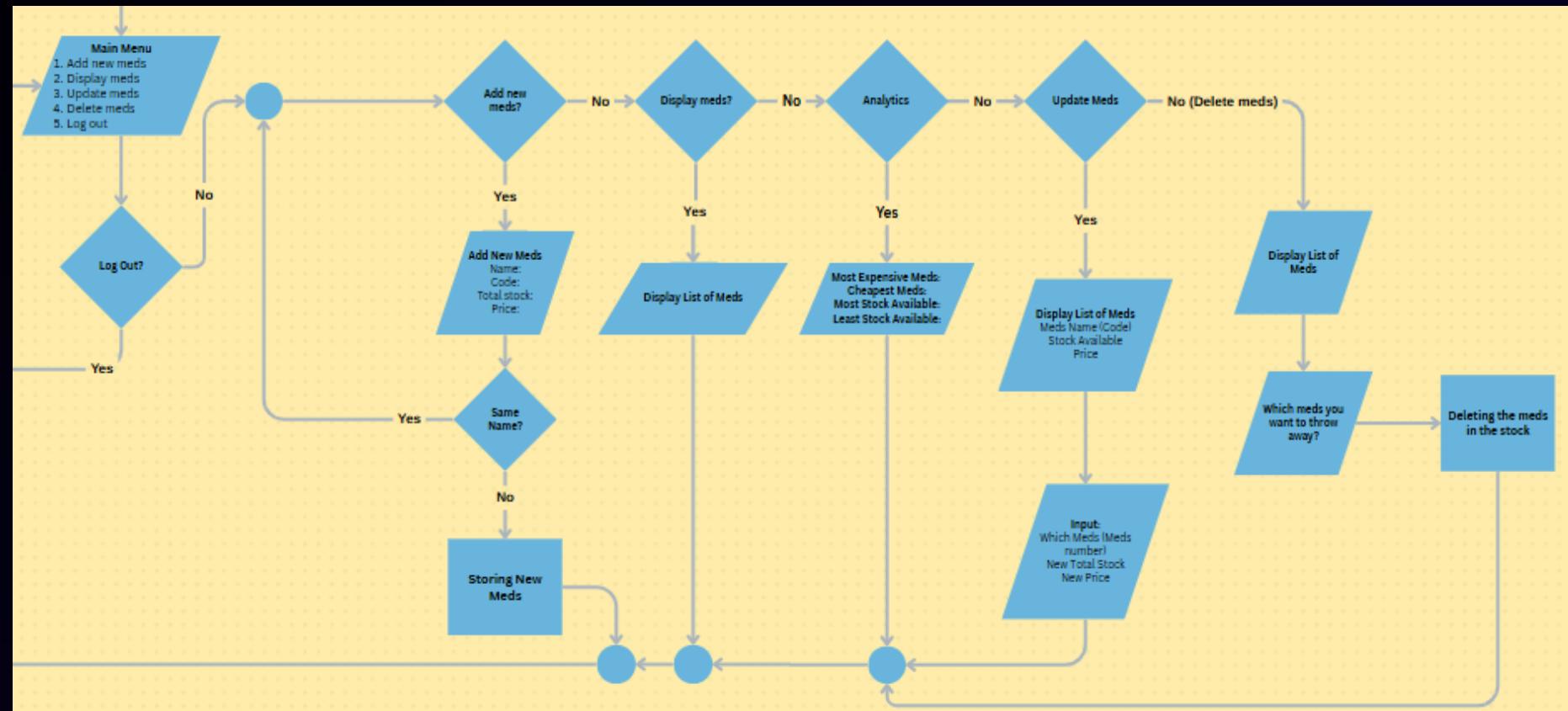
    printf("Is Manager? (1=Yes, 0=No): "); scanf("%d", &u.isManager); getchar();
    FILE *file = fopen("users.txt", "ab");
    fwrite(&u, sizeof(User), 1, file);
    fclose(file);
    printf("Registered. Press Enter..."); getchar(); clearScreen();
}
  
```



User Authorization

Auth.h

menu.h



Manager & Analytics Menu

```

void analyticsMenu() {
    Med meds[100];
    int n = loadMeds(meds);

    if (n == 0) {
        printf("No meds available.\n");
        return;
    }

    int maxPriceIdx = 0, minPriceIdx = 0;
    int maxStockIdx = 0, minStockIdx = 0;

    for (int i = 1; i < n; i++) {
        if (meds[i].price > meds[maxPriceIdx].price) maxPriceIdx = i;
        if (meds[i].price < meds[minPriceIdx].price) minPriceIdx = i;
        if (meds[i].stock > meds[maxStockIdx].stock) maxStockIdx = i;
        if (meds[i].stock < meds[minStockIdx].stock) minStockIdx = i;
    }

    printf("== Analytics ==\n");
    printf("Most Expensive: %s (%.2f)\n", meds[maxPriceIdx].name, meds[maxPriceIdx].price);
    printf("Cheapest : %s (%.2f)\n", meds[minPriceIdx].name, meds[minPriceIdx].price);
    printf("Most Stocks : %s (%d pcs)\n", meds[maxStockIdx].name, meds[maxStockIdx].stock);
    printf("Least Stocks : %s (%d pcs)\n", meds[minStockIdx].name, meds[minStockIdx].stock);
    printf("Press Enter to continue..."); getchar();
}
    
```

Declare library and functions

```

#ifndef MENU_H
#define MENU_H
#include "auth.h"
#include "meds.h"
#include "utils.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
    
```

```

void cashierMenu();
void managerMenu();
void analyticsMenu();
    
```

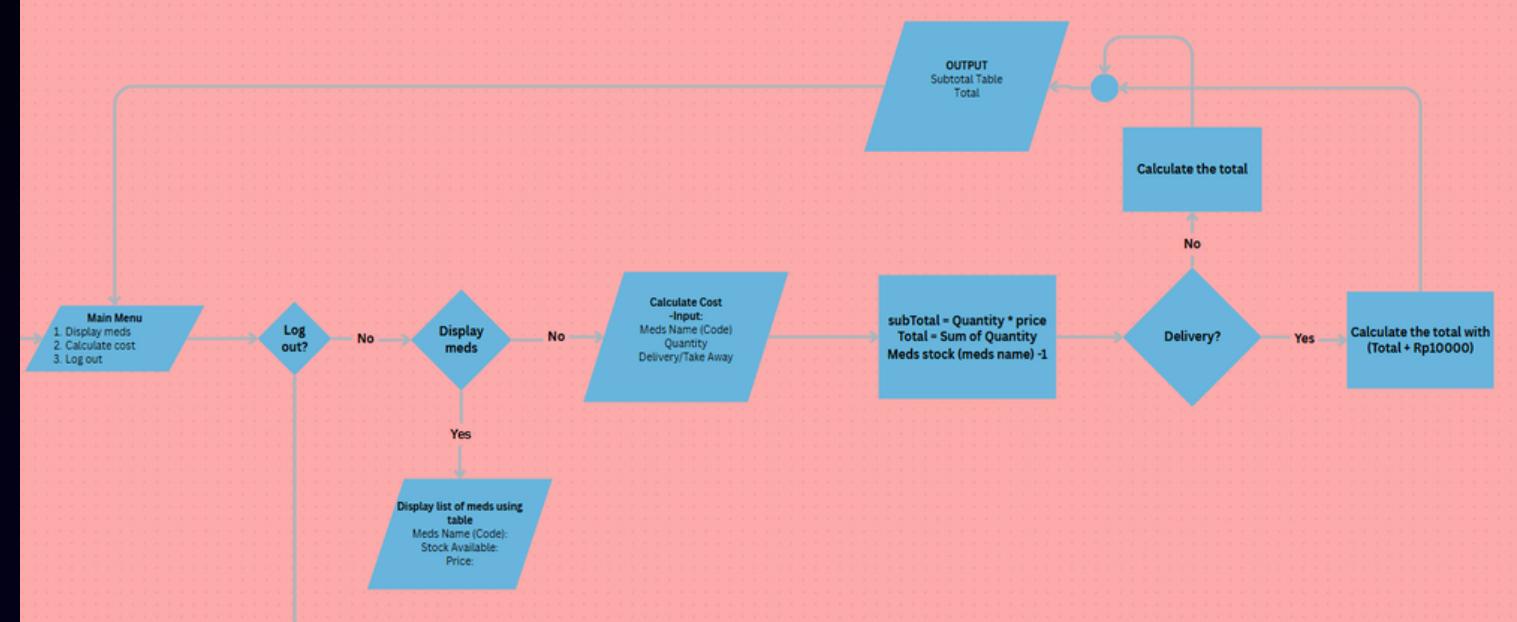
```

void managerMenu() {
    int choice;
    do {
        printf("== Manager Menu ==\n1. Add meds\n2. Display meds\n3. Update meds\n4. Delete meds\n5. Analytics\n6. Log out\nChoice: ");
        scanf("%d", &choice); getchar(); clearScreen();

        switch (choice) {
            case 1: addMed(); break;
            case 2: displayMeds(); break;
            case 3: displayMeds(); updateMed(); break;
            case 4: deleteMed(); break;
            case 5: analyticsMenu(); break;
        }
    } while (choice != 6);
}

#endif
    
```

menu.h



```

void cashierMenu() {
    int choice;
    do {
        printf("== Cashier Menu ==\n1. Display meds\n2. Calculate cost\n3. Log out\nChoice: ");
        scanf("%d", &choice); getchar(); clearScreen();

        if (choice == 1) {
            displayMeds();
        } else if (choice == 2) {
            Med meds[100];
            int n = loadMeds(meds), qty;
            char code[10];
            float total = 0;

            displayMeds();
            while (1) {
                printf("Enter med code (or 'done'): "); gets(code);
                if (strcmp(code, "done") == 0) break;
                for (int i = 0; i < n; i++) {
                    if (strcmp(meds[i].code, code) == 0) {
                        printf("Quantity: "); scanf("%d", &qty); getchar();
                        if (qty <= meds[i].stock) {
                            total += meds[i].price * qty;
                            meds[i].stock -= qty;
                        } else {
                            printf("Not enough stock!\n");
                        }
                    }
                }
                break;
            }

            char deliv;
            printf("Delivery? (y/n): "); scanf("%c", &deliv); getchar();
            if (deliv == 'y') total += 10000;

            printf("TOTAL = Rp%.2f\n", total);
            saveAllMeds(meds, n);
            printf("Press Enter..."); getchar(); clearScreen();
        }
    } while (choice != 3);
}
  
```

Pemrograman C

Background

Objectives

Limitation

User

Flowchart & Code

```
#ifndef MEDS_H
#define MEDS_H
#include "utils.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

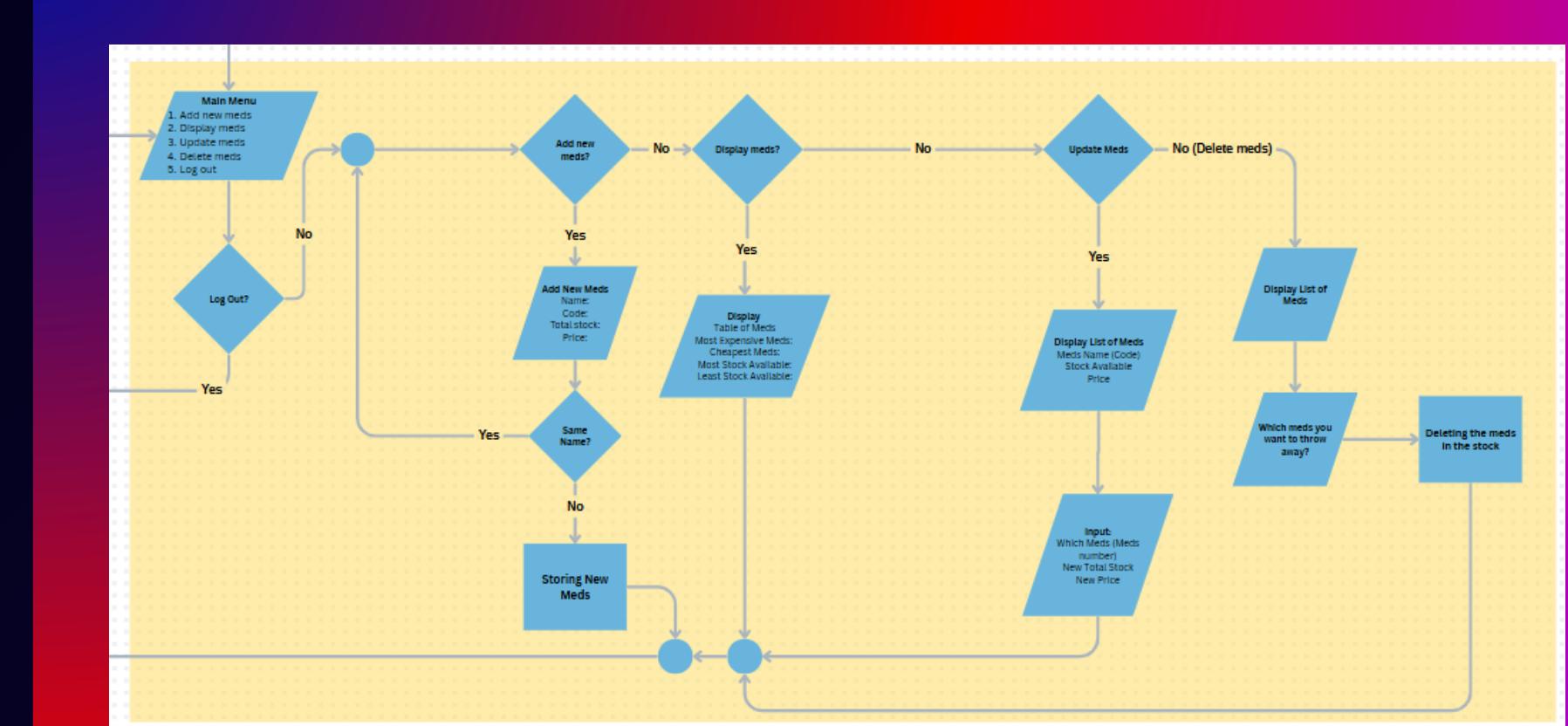
typedef struct { //tipe data sebagai struktur dari sebuah objek obat
    char name[50];
    char code[10];
    int stock;
    float price;
} Med;

int loadMeds(Med meds[]) {
    FILE *file = fopen("meds.txt", "rb");
    int count = 0;
    if (file) {
        while (fread(&meds[count], sizeof(Med), 1, file)) count++;
        fclose(file);
    }
    return count;
}

void storeMed(Med m) {
    FILE *file = fopen("meds.txt", "ab");
    fwrite(&m, sizeof(Med), 1, file);
    fclose(file);
}

void displayMeds() {
    Med m;
    FILE *file = fopen("meds.txt", "rb");
    if (!file) {
        printf("No meds found.\n");
    } else {
        printf("== List of Meds ==\n");
        while (fread(&m, sizeof(Med), 1, file)) {
            printf("%s (%s) - Stock: %d - Rp%.2f\n", m.name, m.code, m.stock, m.price);
        }
        fclose(file);
    }
    printf("Press Enter..."); getchar();
}
}
```

Meds.h



```

int isMedNameExist(const char* name, Med meds[], int n) {
    for (int i = 0; i < n; i++) {
        if (strcmp(meds[i].name, name) == 0) {
            return 1;
        }
    }
    return 0;
}

void addMed() {
    Med m;
    Med meds[100];
    int n = loadMeds(meds);

    printf("Name: ");
    gets(m.name);

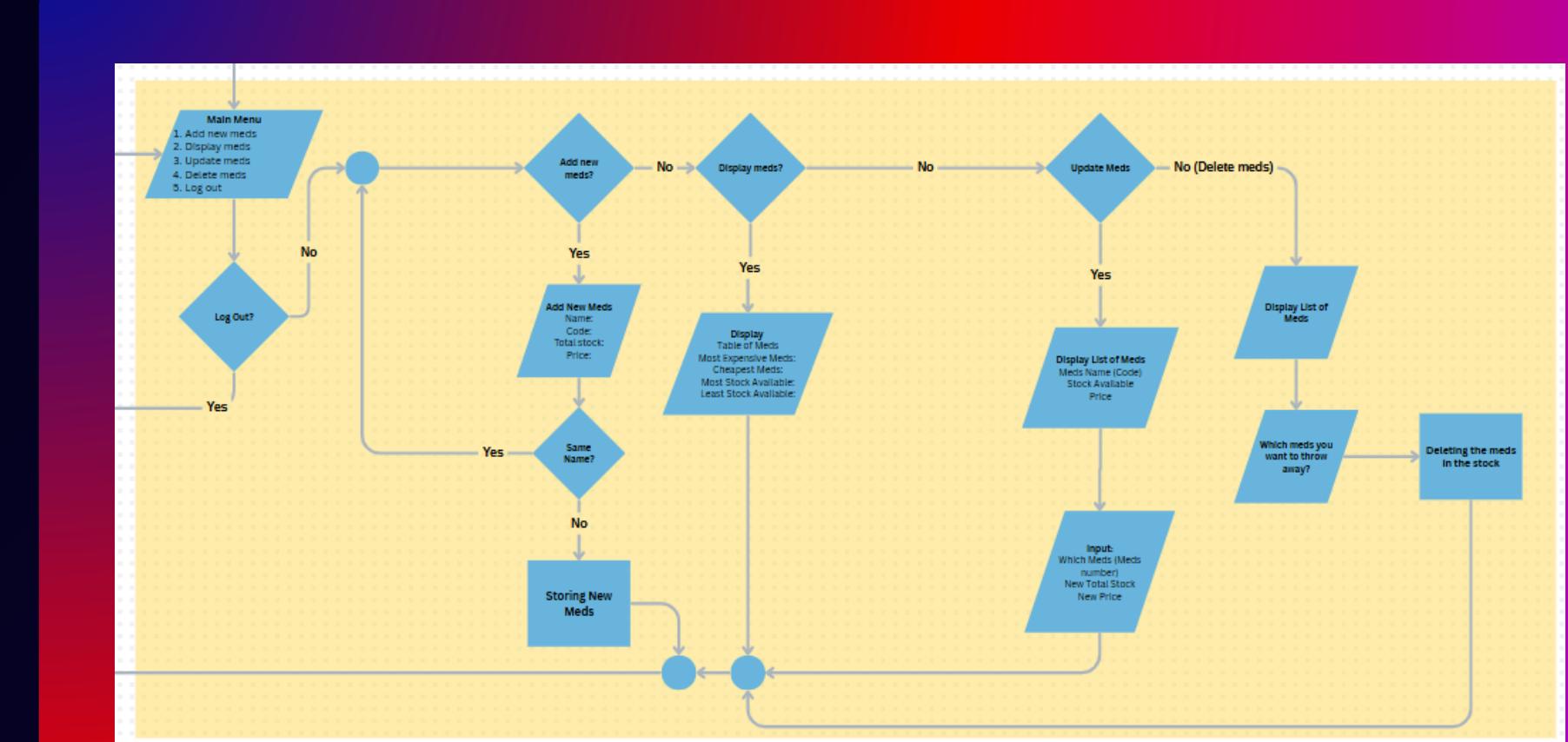
    if (isMedNameExist(m.name, meds)) {
        printf("Error: Med with the same name already exists!\n");
        printf("Press Enter..."); getchar(); clearScreen();
        return;
    }

    printf("Code: "); gets(m.code);
    printf("Stock: "); scanf("%d", &m.stock);
    printf("Price: "); scanf("%f", &m.price);
    getchar();
    storeMed(m);
    printf("Added. Press Enter..."); getchar(); clearScreen();
}

void saveAllMeds(Med meds[], int n) {
    FILE *file = fopen("meds.txt", "wb");
    fwrite(meds, sizeof(Med), n, file);
    fclose(file);
}

```

Meds.h



```

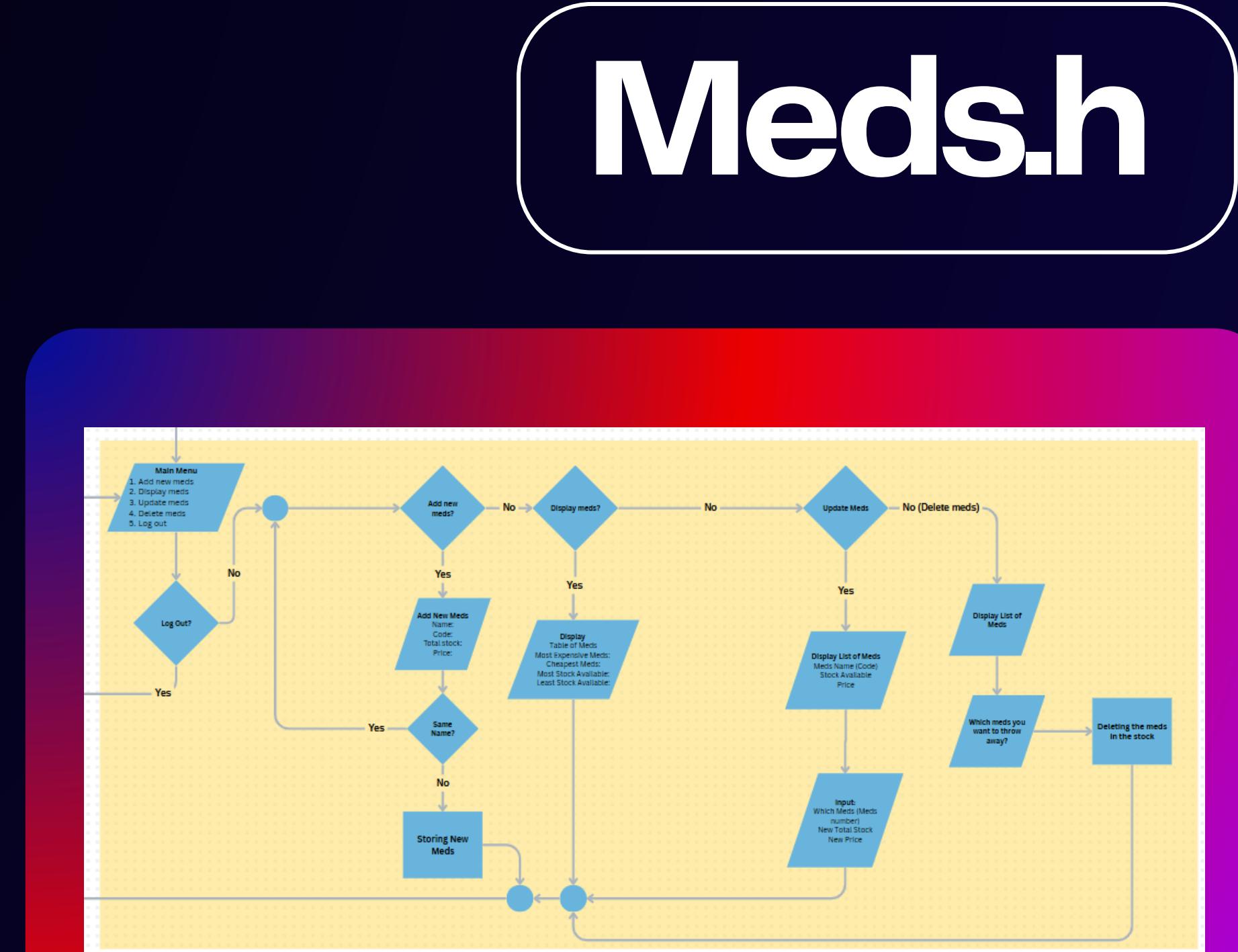
void saveAllMeds(Med meds[], int n) {
    FILE *file = fopen("meds.txt", "wb");
    fwrite(meds, sizeof(Med), n, file);
    fclose(file);
}

void updateMed() {
    Med meds[100];
    int n = loadMeds(meds);
    char code[10];
    printf("Enter med code to update: "); gets(code);
    for (int i = 0; i < n; i++) {
        if (strcmp(meds[i].code, code) == 0) {
            printf("New name: "); gets(meds[i].name);
            printf("New stock: "); scanf("%d", &meds[i].stock);
            printf("New price: "); scanf("%f", &meds[i].price); getchar();
            break;
        }
    }
    saveAllMeds(meds, n);
    printf("Updated. Press Enter..."); getchar(); clearScreen();
}

void deleteMed() {
    Med meds[100];
    int n = loadMeds(meds);
    char code[10];
    displayMeds();
    printf("Enter med code to delete: "); gets(code);
    for (int i = 0; i < n; i++) {
        if (strcmp(meds[i].code, code) == 0) {
            for (int j = i; j < n - 1; j++) meds[j] = meds[j + 1];
            n--;
            break;
        }
    }
    saveAllMeds(meds, n);
    printf("Deleted. Press Enter..."); getchar(); clearScreen();
}

#endif

```





BONUS

Utilsh

```
#ifndef UTILS_H
#define UTILS_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void clearScreen() {
#ifndef _WIN32
    system("cls");
#else
    system("clear");
#endif
}

#endif
```

Thank You!

-PharmaSys by Group 2