#include <iostream>

#include <fstream>

#include <list>

#include <iomanip>

#include <string>

using namespace std;

struct Patient {

string id;

string name;

string severity;

int waitTime;

string status;

};

void loadPatients(list<Patient>& patients);

void savePatients(const list<Patient>& patients);

void registerPatient(list<Patient>& patients);

void displayQueue(const list<Patient>& patients);

void admitPatient(list<Patient>& patients);

void viewSummary(const list<Patient>& patients);

int main() {

list<Patient> patients;

loadPatients(patients);

int choice;

do {

cout << "\n\*\*\*\*\*\* Welcome to Emergency Room Queue Manager \*\*\*\*\*\*\n";

cout << "1. Register New Patient\n2. Display Patient Queue\n3. Admit Patient\n";

cout << "4. View ER Summary\n5. Exit\n→ ";

cin >> choice;

cin.ignore();

switch (choice) {

case 1: registerPatient(patients); break;

case 2: displayQueue(patients); break;

case 3: admitPatient(patients); break;

case 4: viewSummary(patients); break;

case 5:

savePatients(patients);

cout << "Patient queue saved to patients.txt. Have a safe shift!\n";

break;

default:

cout << "Invalid option. Try again.\n";

}

} while (choice != 5);

return 0;

}

void loadPatients(list<Patient>& patients) {

ifstream file("patients.txt");

if (!file) return;

Patient p;

char comma;

while (file >> p.id >> comma >> p.name >> comma >> p.severity >> comma >> p.waitTime >> comma >> p.status) {

patients.push\_back(p);

}

file.close();

}

void savePatients(const list<Patient>& patients) {

ofstream file("patients.txt");

for (const auto& p : patients) {

file << p.id << "," << p.name << "," << p.severity << ","

<< p.waitTime << "," << p.status << "\n";

}

file.close();

}

void registerPatient(list<Patient>& patients) {

Patient p;

cout << "Enter patient ID: ";

cin >> p.id;

cin.ignore();

cout << "Enter name: ";

getline(cin, p.name);

cout << "Enter condition severity (Low/Medium/High/Critical): ";

cin >> p.severity;

cout << "Enter estimated wait time (in minutes): ";

cin >> p.waitTime;

p.status = "Waiting";

patients.push\_back(p);

cout << "Patient registered successfully!\n";

}

void displayQueue(const list<Patient>& patients) {

if (patients.empty()) {

cout << "No patients currently in queue.\n";

return;

}

cout << "\n=========== ER WAITING LIST ===========\n";

cout << left << setw(8) << "ID" << setw(20) << "Name" << setw(12)

<< "Severity" << setw(10) << "Wait(min)" << "Status\n";

cout << "------------------------------------------------------\n";

for (const auto& p : patients) {

cout << left << setw(8) << p.id << setw(20) << p.name

<< setw(12) << p.severity << setw(10) << p.waitTime << p.status << "\n";

}

}

void admitPatient(list<Patient>& patients) {

if (patients.empty()) {

cout << "No patients to admit.\n";

return;

}

for (auto& p : patients) {

if (p.status == "Waiting") {

p.status = "Admitted";

cout << "Patient " << p.id << " has been admitted to the ER.\n";

return;

}

}

cout << "All patients have already been admitted.\n";

}

void viewSummary(const list<Patient>& patients) {

int total = 0, admitted = 0, waiting = 0;

float totalWait = 0;

for (const auto& p : patients) {

total++;

if (p.status == "Admitted") admitted++;

else {

waiting++;

totalWait += p.waitTime;

}

}

float avgWait = (waiting > 0) ? totalWait / waiting : 0;

cout << "\n========== ER SUMMARY ==========\n";

cout << "Total Patients: " << total << "\n";

cout << "Admitted: " << admitted << "\n";

cout << "Waiting: " << waiting << "\n";

cout << "Avg Wait Time (Waiting): " << fixed << setprecision(2) << avgWait << " minutes\n";

cout << "================================\n";

}