import java.util.\*;

public class CollegeAdmissionSystem {

// User Class (Parent Class for Applicant, Admin, Cashier, etc.)

public abstract static class User {

protected String id;

protected String password;

protected String role;

public User(String id, String password, String role) {

this.id = id;

this.password = password;

this.role = role;

}

public abstract void login(String id, String password);

}

// Applicant Class

public static class Applicant extends User {

private String name;

private String qualification;

private String applicationStatus;

private double dues;

public Applicant(String id, String password, String name, String qualification) {

super(id, password, "Applicant");

this.name = name;

this.qualification = qualification;

this.applicationStatus = "Drafted";

this.dues = 1000.0; // Initial dues set to 1000

}

@Override

public void login(String id, String password) {

if (this.id.equals(id) && this.password.equals(password)) {

System.out.println("Applicant logged in successfully.");

} else {

System.out.println("Invalid login credentials.");

}

}

public void submitApplication() {

this.applicationStatus = "Submitted";

System.out.println("Application submitted successfully.");

}

public void trackApplicationStatus() {

System.out.println("Application Status: " + applicationStatus);

}

public void updateApplication(String additionalInfo) {

this.applicationStatus = "UnderReview";

System.out.println("Application updated with additional information: " + additionalInfo);

}

public void becomeStudent() {

this.role = "Student";

System.out.println("Applicant is now a Student.");

}

public void payDues(double amount) {

if (dues > 0) {

dues -= amount;

System.out.println("Payment successful. Remaining dues: " + dues);

} else {

System.out.println("No dues remaining.");

}

}

public double getDues() {

return dues;

}

}

// Admin Class

public static class Admin extends User {

private List<Applicant> students = new ArrayList<>();

private List<Cashier> cashiers = new ArrayList<>();

private List<User> visitors = new ArrayList<>();

public Admin(String id, String password) {

super(id, password, "Admin");

}

@Override

public void login(String id, String password) {

if (this.id.equals(id) && this.password.equals(password)) {

System.out.println("Admin logged in successfully.");

} else {

System.out.println("Invalid login credentials.");

}

}

public void reviewApplication(Applicant applicant) {

System.out.println("Reviewing application for: " + applicant.name);

applicant.updateApplication("Requested missing information.");

}

public void approveApplication(Applicant applicant) {

applicant.submitApplication();

applicant.becomeStudent();

students.add(applicant);

System.out.println("Application approved for: " + applicant.name);

}

public void viewApplicantList(List<Applicant> applicants) {

System.out.println("\nApplicant List:");

for (Applicant applicant : applicants) {

System.out.println("ID: " + applicant.id + ", Name: " + applicant.name + ", Status: " + applicant.applicationStatus);

}

}

public void viewStudentList() {

System.out.println("\nStudent List and Payment Dues:");

for (Applicant student : students) {

System.out.println("ID: " + student.id + ", Name: " + student.name + ", Dues: " + student.getDues());

}

}

public void viewCashierList() {

System.out.println("\nCashier List:");

for (Cashier cashier : cashiers) {

System.out.println("ID: " + cashier.id + ", Name: " + cashier.name);

}

}

public void viewVisitorList() {

System.out.println("\nVisitor List:");

for (User visitor : visitors) {

System.out.println("ID: " + visitor.id + ", Name: " + visitor.role);

}

}

public void addVisitor(User visitor) {

visitors.add(visitor);

System.out.println("Visitor added: " + visitor.id);

}

}

// Cashier Class

public static class Cashier extends User {

private String name;

public Cashier(String id, String password, String name) {

super(id, password, "Cashier");

this.name = name;

}

@Override

public void login(String id, String password) {

if (this.id.equals(id) && this.password.equals(password)) {

System.out.println("Cashier logged in successfully.");

} else {

System.out.println("Invalid login credentials.");

}

}

public void clearPayment(Applicant student) {

if (student.getDues() > 0) {

student.payDues(student.getDues()); // Clear the dues

System.out.println("Payment cleared for " + student.name);

} else {

System.out.println("No dues to clear for " + student.name);

}

}

}

// Visitor Class (can be a generic user, no role-specific functionality)

public static class Visitor extends User {

public Visitor(String id) {

super(id, "", "Visitor");

}

@Override

public void login(String id, String password) {

System.out.println("Visitor logged in.");

}

}

// Main Class to Simulate System Behavior

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Initialize admin user

Admin admin = new Admin("admin123", "admin123");

// Lists for Applicants, Students, Cashiers, and Visitors

List<Applicant> applicants = new ArrayList<>();

List<Applicant> students = admin.students;

List<Cashier> cashiers = admin.cashiers;

List<User> allUsers = new ArrayList<>();

List<User> visitors = admin.visitors;

allUsers.add(admin); // Initially adding admin

visitors.add(new Visitor("visitor1")); // Add a visitor

// Display menu

while (true) {

System.out.println("\n--- College Admission Management System ---");

System.out.println("1. Register as Applicant");

System.out.println("2. Register as Admin");

System.out.println("3. Register as Cashier");

System.out.println("4. Register as Visitor");

System.out.println("5. Applicant Actions");

System.out.println("6. Admin Actions");

System.out.println("7. Cashier Actions");

System.out.println("8. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

switch (choice) {

case 1:

// Register as Applicant

System.out.print("Enter Access Key: ");

String accessKey = scanner.nextLine();

if (!accessKey.equals("12311037")) {

System.out.println("Invalid Access Key. Cannot register.");

break;

}

System.out.print("Enter Applicant ID: ");

String appId = scanner.nextLine();

System.out.print("Enter Applicant Password: ");

String appPassword = scanner.nextLine();

System.out.print("Enter Applicant Name: ");

String appName = scanner.nextLine();

System.out.print("Enter Applicant Qualification: ");

String appQualification = scanner.nextLine();

Applicant newApplicant = new Applicant(appId, appPassword, appName, appQualification);

applicants.add(newApplicant);

allUsers.add(newApplicant); // Add applicant to all users list

System.out.println("Applicant registered successfully.");

break;

case 2:

// Register as Admin

System.out.print("Enter Access Key: ");

String adminAccessKey = scanner.nextLine();

if (!adminAccessKey.equals("12311037")) {

System.out.println("Invalid Access Key. Cannot register.");

break;

}

System.out.print("Enter Admin ID: ");

String adminId = scanner.nextLine();

System.out.print("Enter Admin Password: ");

String adminPassword = scanner.nextLine();

admin = new Admin(adminId, adminPassword);

allUsers.add(admin); // Add admin to all users list

System.out.println("Admin registered successfully.");

break;

case 3:

// Register as Cashier

System.out.print("Enter Access Key: ");

String cashierAccessKey = scanner.nextLine();

if (!cashierAccessKey.equals("12311038")) {

System.out.println("Invalid Access Key. Cannot register.");

break;

}

System.out.print("Enter Cashier ID: ");

String cashierId = scanner.nextLine();

System.out.print("Enter Cashier Password: ");

String cashierPassword = scanner.nextLine();

System.out.print("Enter Cashier Name: ");

String cashierName = scanner.nextLine();

Cashier cashier = new Cashier(cashierId, cashierPassword, cashierName);

cashiers.add(cashier);

allUsers.add(cashier);

System.out.println("Cashier registered successfully.");

break;

case 4:

// Register as Visitor

System.out.print("Enter Visitor ID: ");

String visitorId = scanner.nextLine();

Visitor visitor = new Visitor(visitorId);

visitors.add(visitor);

System.out.println("Visitor registered successfully.");

break;

case 5:

// Applicant Actions

System.out.print("Enter Applicant ID for login: ");

String applicantId = scanner.nextLine();

System.out.print("Enter Password: ");

String applicantLoginPassword = scanner.nextLine();

// Find applicant

Applicant applicantToLogin = null;

for (User user : allUsers) {

if (user instanceof Applicant && user.id.equals(applicantId)) {

applicantToLogin = (Applicant) user;

break;

}

}

if (applicantToLogin != null) {

applicantToLogin.login(applicantId, applicantLoginPassword);

System.out.println("1. Submit Application");

System.out.println("2. Track Application Status");

System.out.print("Choose an action: ");

int applicantAction = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (applicantAction == 1) {

applicantToLogin.submitApplication();

} else if (applicantAction == 2) {

applicantToLogin.trackApplicationStatus();

}

} else {

System.out.println("Applicant not found.");

}

break;

case 6:

// Admin Actions

System.out.print("Enter Admin ID for login: ");

String adminLoginId = scanner.nextLine();

System.out.print("Enter Password: ");

String adminLoginPassword = scanner.nextLine();

if (admin.id.equals(adminLoginId) && admin.password.equals(adminLoginPassword)) {

System.out.println("Admin logged in successfully.");

System.out.println("1. View Applicant List");

System.out.println("2. Approve Applicant");

System.out.println("3. View Student List");

System.out.println("4. View Cashier List");

System.out.println("5. View Visitor List");

System.out.print("Choose an action: ");

int adminAction = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (adminAction == 1) {

admin.viewApplicantList(applicants);

} else if (adminAction == 2) {

System.out.print("Enter applicant ID to approve: ");

String applicantIdToApprove = scanner.nextLine();

for (Applicant applicant : applicants) {

if (applicant.id.equals(applicantIdToApprove)) {

admin.approveApplication(applicant);

students.add(applicant); // Move approved applicants to students list

applicants.remove(applicant);

break;

}

}

} else if (adminAction == 3) {

admin.viewStudentList();

} else if (adminAction == 4) {

admin.viewCashierList();

} else if (adminAction == 5) {

admin.viewVisitorList();

}

} else {

System.out.println("Invalid admin login.");

}

break;

case 7:

// Cashier Actions

System.out.print("Enter Cashier ID for login: ");

String cashierLoginId = scanner.nextLine();

System.out.print("Enter Password: ");

String cashierLoginPassword = scanner.nextLine();

// Find cashier and process payments

Cashier cashierToLogin = null;

for (User user : allUsers) {

if (user instanceof Cashier && user.id.equals(cashierLoginId)) {

cashierToLogin = (Cashier) user;

break;

}

}

if (cashierToLogin != null) {

cashierToLogin.login(cashierLoginId, cashierLoginPassword);

System.out.println("1. Clear Payment for Student");

System.out.print("Choose an action: ");

int cashierAction = scanner.nextInt();

scanner.nextLine(); // Consume newline

if (cashierAction == 1) {

System.out.print("Enter student ID to clear payment: ");

String studentIdForPayment = scanner.nextLine();

for (Applicant student : students) {

if (student.id.equals(studentIdForPayment)) {

cashierToLogin.clearPayment(student);

break;

}

}

}

} else {

System.out.println("Cashier not found.");

}

break;

case 8:

System.out.println("Exiting the system...");

scanner.close();

return;

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

}