SUNDAY LECTURE:

1.Java project (Swiggy)

2.src

3.default packages(should not use anything inside this)

4.models (java classes)

a.

b.

c.

d.

e.

5.ui (main class)

6.repository (repo class for all the models class)

a.

b.

c.

d.

e.

There are 2 methods to accomplish this project:

1. Storing the data in the list inside the repository of the respective classes and retrieving it using it in the main class, where the main class, the list should be created and the userRepo list fill be stored overhere and the it will be retrieved/printed accordingly.
2. Storing in the database and retrieving it using the respective Repo class of each classes

MAIN CLASS:

package ui;

import java.util.List;

import models.User;

import repository.UserRepo;

public class Main {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

UserRepo userRepo = new UserRepo();

List<User> users = userRepo.getAllUsers();

for (User u : users) {

System.***out***.println("Name : "+u.Name+"\t"+"User Id Type : "+u.UserTypeId);

// System.out.println();

}

}

}

USER CLASS:

package models;

public class User {

public int UserId;

public String Name;

public int UserTypeId;

public long PhoneNumber;

}

USER REPO CLASS:

package repository;

import models.User;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

//import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class UserRepo {

private Connection connect() {

Connection conn = null;

try {

// Class.forName("com.mysql.cj.jdbc.Driver");

String url = "jdbc:mysql://localhost:3306/swiggy";

String user = "root"; // Replace with your MySQL username

String password = "root"; // Replace with your MySQL password

conn = DriverManager.*getConnection*(url, user, password);

} catch (Exception e) {

e.printStackTrace();

}

return conn;

}

// Method to get a user by ID

public User getUserById(int userId) {

User user = null;

String query = "SELECT \* FROM User WHERE UserId = ?";

try (Connection conn = connect();

PreparedStatement pstmt = conn.prepareStatement(query)) {

pstmt.setInt(1, userId);

ResultSet rs = pstmt.executeQuery();

if (rs.next()) {

user.Name = rs.getString("Name");

user.UserTypeId = rs.getInt("UserTypeId");

user.PhoneNumber = rs.getLong("PhoneNumber");

}

} catch (Exception e) {

e.printStackTrace();

}

return user;

}

// Method to get all users

public List<User> getAllUsers() {

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

String query = "SELECT \* FROM User";

try (Connection conn = connect();

PreparedStatement pstmt = conn.prepareStatement(query);

ResultSet rs = pstmt.executeQuery()) {

while (rs.next()) {

User user = new User();

user.UserId = rs.getInt("UserId");

user.Name = rs.getString("Name");

user.UserTypeId = rs.getInt("UserTypeId");

user.PhoneNumber = rs.getLong("PhoneNumber");

users.add(user);

}

} catch (Exception e) {

e.printStackTrace();

}

return users;

}

}

NEED CONCENTRATION ON THE LISTS:



Enter user

store in user obj => userRepo.UserRegister(User user)

userRepo.getAllUSer()

{

list of user fetch

}

userRepo.GetByID(int id)

{

single user return

}

userRepo.UserRegistration(User user)

{

insert logic

}

**COMPLETED**

orderRepo.GetOrderById(int id)

{

single return

}

orderRepo.GetorderByUSerId(int id)

{

return order

}

UI Package

{

list of user obj = userRepo.getAllUSer()

for loop - print

}

1.UserReg

2.ShowUSers

3 .place

enter name, address,

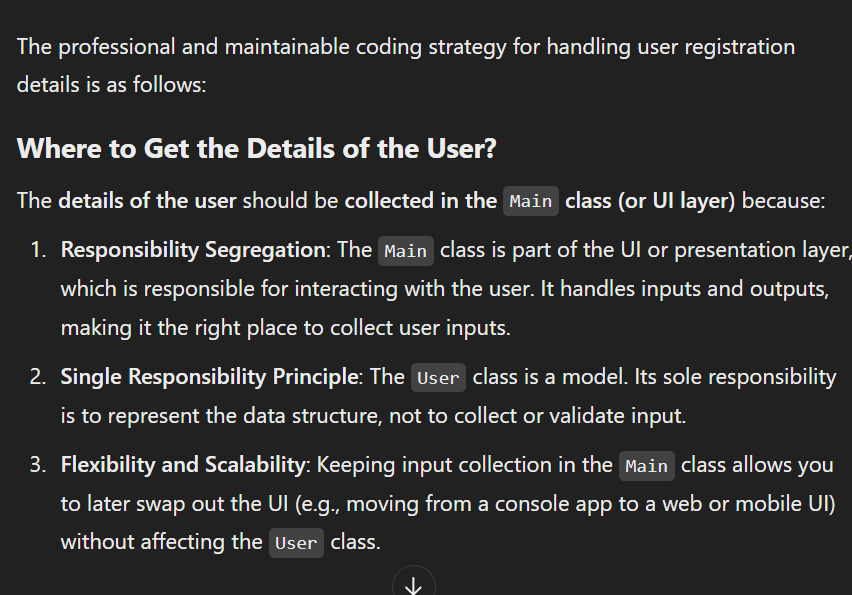
User user = new User();

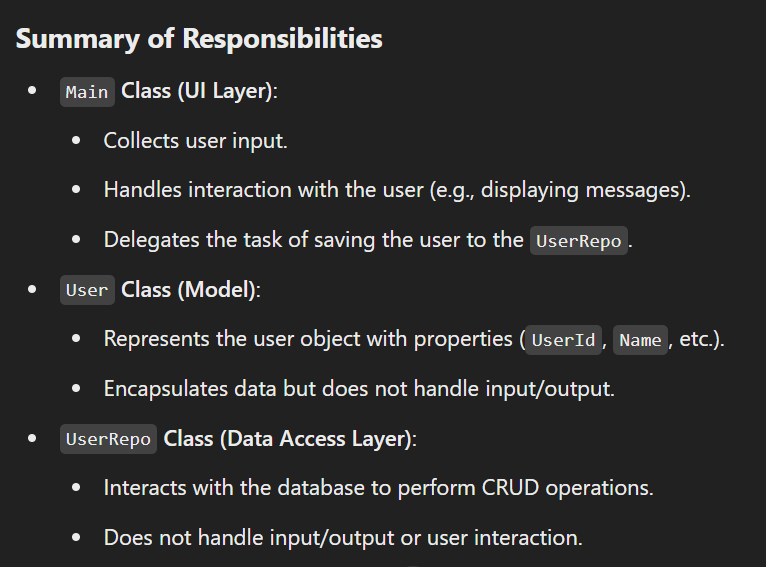
user.name = "Naveen"

user.userType = "1"

userRepo.UserRegistration(user);

**answer for my doubt!**





**Food class -> declaration of attributes**

**FoodRepo class -> database storage**

**Main class -> for funtions?**

**Particular method should be declared in which class?**

**Doubt?**

**SUNDAY LECTURE (17/11/2024)**

**so the answer for my doubt is…**

**the models**

**should only contain the attributes**

**the repo**

**should only contain the db storing and retrieving codes**

**the ui**

**which contains main class**

**all the other methods of the respective classes**

**can be created a new class with the respective names and the methods should be updated into that**

**( this is one of the works )**

ENTITY DIAGRAM

FLOW DIAGRAM SEE EVERY DAY 15 MINS

SYSTEM DIAGRAM

(CRUCIAL STEP TO ATTRACT THE MONITORING PEOPLE)

HACKER RANK (PRACTICE FOR APTITUDE)

SINGLE RESPONSIBILITY PRINCIPLE

(SHOULD NOT BE VOILATED)

int currentUser = 0

int selectAddress = 0

user - address

user insert

last userid

private AddAddress(userID, s.....)

{

}

adduser,

showuser

------------

1.Login

2.Register

-----

1.ShowFood

2.ShowAddrsses

3.AddAddress

1

2

3

press 0 once you done with orderedItems

0

userRepo.ShowAllAddress()

select address

selectedAdd = 2

orderRepo.PLaceOrder()

selectAddress = 0

Thank you

userID

Enter ur userID: 1

userID(1),

currentUserId = id

Hi Naveen,

currentId =0;

show foods

list orderedItems

orderRepo.PLaceOrder(currentUseruser, selectAddress, ListofItem)

generatedOrderID = orders (userid, addressId)

for(ListofItem)

orderItmes(orderID, ListofItem.quanty, food)

Main.java

package ui;

import java.util.\*;

import java.util.Scanner;

import models.User;

import repository.UserRepo;

import models.Food;

import repository.FoodRepo;

public class Main {

public static void main(String[] args) {

UserRepo userRepo = new UserRepo();

FoodRepo foodRepo = new FoodRepo();

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("Select an option : ");

System.out.println("1 : Login");

System.out.println("2 : Register");

System.out.println("Select an option : ");

System.out.println("1 : Add user");

System.out.println("2 : Show user");

System.out.println("3 : Place order");

System.out.println("4 : Add food");

int choice = scanner.nextInt();

switch (choice) {

case 1:

User newUser = collectUserDetails(scanner);

// sir's line

User addedUser = userRepo.AddUser(newUser);

if (addedUser != null) {

System.out.println("User added Successfully: "+addedUser.Name);

}

else {

System.out.println("Failed to add user");

}

break;

case 2:

showUser(userRepo,scanner);

break;

case 3:

// show foods

List<Food> foods = foodRepo.showAllFoods();

System.out.println("Today's Menu");

for (Food f : foods) {

System.out.println("Food ID : "+f.FoodId+"\t"+"Name : "+f.Name);

}

// get orders

System.out.println("Press 0 to complete your order");

System.out.println("Enter Food Id and Quantity :");

Map<Integer,Integer> order = new HashMap<>();

while (true) {

int foodId = scanner.nextInt();

int quantity = scanner.nextInt();

if (foodId == 0) {

break;

}

else {

order.put(foodId,quantity);

}

}

// show orders

System.out.println("Your Orders : ");

for (Map.Entry<Integer,Integer> entry : order.entrySet()) {

int foodId2 = entry.getKey();

int quantity2 = entry.getValue();

Food orderedFood = foods.stream()

.filter(f -> f.FoodId == foodId2)

.findFirst()

.orElse(null);

if (orderedFood != null) {

System.out.println(orderedFood.Name+"\t"+quantity2);

}

else {

System.out.println("Food Id : "+foodId2+" not found in today's menu");

}

}

break;

case 4:

Food newFood = collectFoodDetails(scanner);

Food addedFood = foodRepo.addNewFood(newFood);

if (addedFood != null) {

System.out.println("Food Added Successfully!");

}

else {

System.out.println("Failed To Add Food");

}

}

}

}

// register a new user -> 1

public static User collectUserDetails(Scanner scanner) {

scanner.nextLine();

System.out.println("Enter Name : ");

String name = scanner.nextLine();

System.out.println("Select User Type : ");

System.out.println("1 : Vendor");

System.out.println("2 : Buyer");

System.out.println("3 : Delivery Person");

int user\_type\_id = scanner.nextInt();

System.out.println("Enter Phone Number");

long phone\_number = scanner.nextLong();

User newUser = new User();

newUser.Name = name;

newUser.UserTypeId = user\_type\_id;

newUser.PhoneNumber = phone\_number;

return newUser;

}

// show user -> 2

public static void showUser(UserRepo userRepo,Scanner scanner) {

System.out.println("Select an option : ");

System.out.println("1 : Display user by ID");

System.out.println("2 : Display all users");

int display\_choice = scanner.nextInt();

switch (display\_choice) {

case 1:

System.out.println("Enter User Id : ");

int user\_id = scanner.nextInt();

User user = userRepo.getUserById(user\_id);

System.out.println("Name : "+user.Name+"\t"+"Phone Number : "+user.PhoneNumber);

break;

case 2:

List<User> users = userRepo.getAllUsers();

for (User u : users) {

System.out.println("Name : "+u.Name+"\t"+"User Id Type : "+u.UserTypeId);

}

break;

}

}

// new food adding

public static Food collectFoodDetails(Scanner scanner) {

scanner.nextLine();

System.out.println("Enter Food Name : ");

String food\_Name = scanner.nextLine();

System.out.println("Quantity : ");

int total\_Quantity = scanner.nextInt();

scanner.nextLine();

System.out.println("Food Type : ");

String food\_Type = scanner.nextLine();

Food newFood = new Food();

newFood.Name = food\_Name;

newFood.Quantity = total\_Quantity;

newFood.FoodType = food\_Type;

return newFood;

}

}

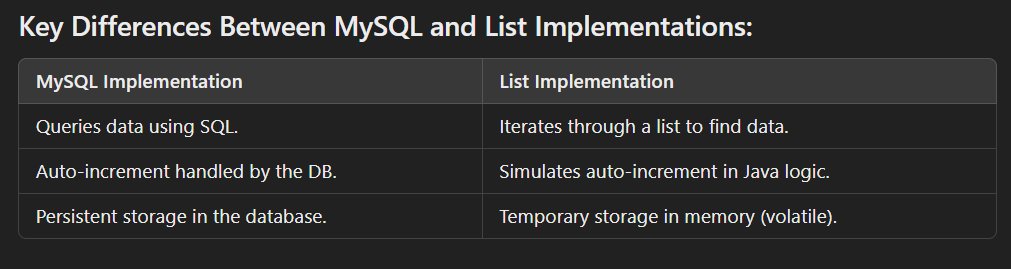
**DOUBT:**

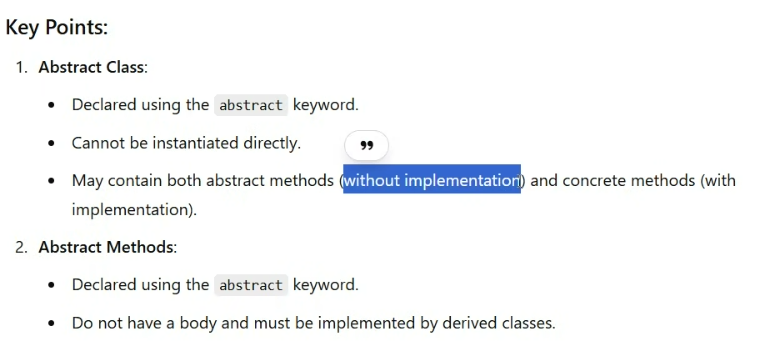
**WHERE TO DECLARE THE REPO REQUEST IN**

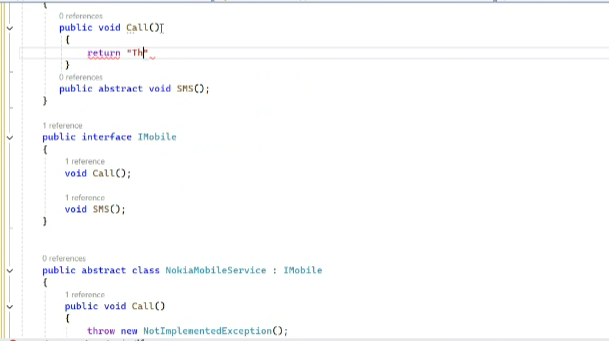
**MAIN OR**

**PARTICLAR UI CLASS**

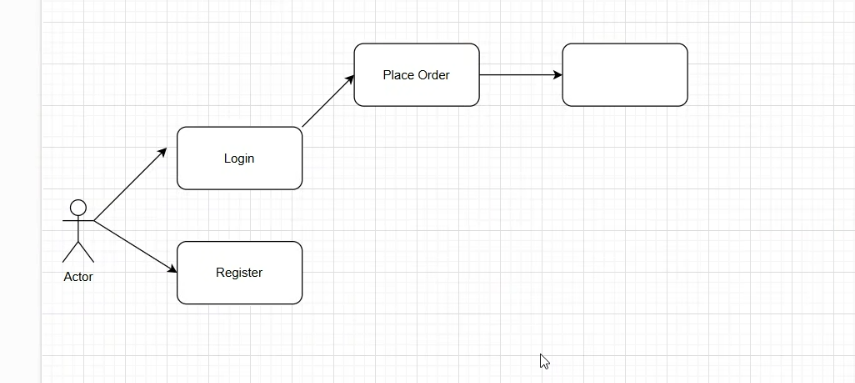
**MYSQL TO LIST IMPLEMENTATION**

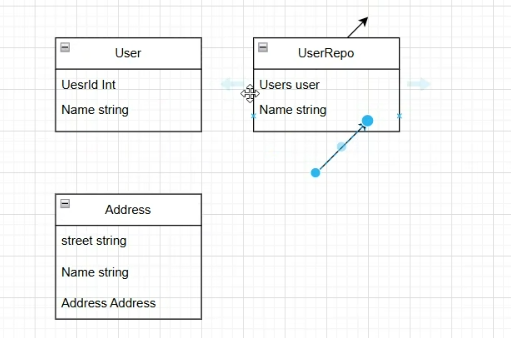
****

****

****

**Flow diagram:**

****

****

**b**