MINI PROJECT 3

NAME : SANJEEV N
DATE : 25-07-25

1 . ADD PACKAGES DETAILS

```
Holiday Package Menu

1. Add Package details
2. Display all package details
3. Search for a package with package id
4. Calculate package cost based on package id
5. Exit
Enter choice: 1
Enter Package Id: PKG1234
Enter Source Place: COIMBATORE
Enter Destination Place: USA
Enter Number of Days: 10
Enter Basic Fare: 25000
Package added successfully.
```

2. DISPLAY ALL PACKAGE DETAILS

```
1. Add Package details
2. Display all package details
3. Search for a package with package id
4. Calculate package cost based on package id
5. Exit
Enter choice: 2
All Packages:
Package [packageId=PKG1234, sourcePlace=COIMBATORE, destinationPlace=USA, noOfDays=10, basicFare=25000.00, packageCost=0.00]
```

3. SEARCH FOR PACKAGES WITH PACKAGE ID

```
1. Add Package details
2. Display all package details
3. Search for a package with package id
4. Calculate package cost based on package id
5. Exit
Enter choice: 3
Enter Package Id to search: PKG1234
Package [packageId=PKG1234, sourcePlace=COIMBATORE, destinationPlace=USA, noOfDays=10, basicFare=25000.00, packageCost=0.00]
```

4.CALCULATE PACKAGE COST BASED ON PACKAGE ID

```
1. Add Package details
2. Display all package details
3. Search for a package with package id
4. Calculate package cost based on package id
5. Exit
Enter choice: 4
Enter Package Id to calculate cost: PKG1234
Cost calculated.
Package Cost for PKG1234 is: 266000.00
```

5.EXIT

```
1. Add Package details
2. Display all package details
3. Search for a package with package id
4. Calculate package cost based on package id
5. Exit
Enter choice: 5
Exiting program.
```

PackageMain.java

```java
package com.client;
import java.util.List;
import java.util.Scanner;
import com.model.Package;
import com.service.PackageService;
import com.service.PackageServiceImpl;
import com.exception.InvalidPackageIdException;
public class PackageMain {
  public static void main(String[] args) {
    PackageService packageService = new PackageServiceImpl();
    Scanner sc = new Scanner(System.in);
    int choice;
    System.out.println("Holiday Package Menu");
    do {
      System.out.println("\n1. Add Package details");
      System.out.println("2. Display all package details");
      System.out.println("3. Search for a package with package id");
      System.out.println("4. Calculate package cost based on package id");
      System.out.println("5. Exit");
      System.out.print("Enter choice: ");
      choice = sc.nextInt();
      sc.nextLine();
      switch (choice) {
        case 1:
          try {
            Package p = new Package();
            System.out.print("Enter Package Id: ");
            p.setPackageId(sc.nextLine());
            System.out.print("Enter Source Place: ");
            p.setSourcePlace(sc.nextLine());
            System.out.print("Enter Destination Place: ");
            p.setDestinationPlace(sc.nextLine());
            System.out.print("Enter Number of Days: ");
            p.setNoOfDays(sc.nextInt());
```

```
                System.out.print("Enter Basic Fare: ");
                p.setBasicFare(sc.nextDouble());
                sc.nextLine();
                packageService.addPackage(p);
                System.out.println("Package added successfully.");
            } catch (InvalidPackageIdException e) {
                System.out.println(e.getMessage());
            }
            break;
        case 2:
            List<Package> list = packageService.fetchAllPackages();
            if (list.isEmpty()) {
                System.out.println("No packages added yet.");
            } else {
                System.out.println("All Packages:");
                list.forEach(pkg -> System.out.println(pkg));
            }
            break;
        case 3:
            System.out.print("Enter Package Id to search: ");
            String searchId = sc.nextLine();
            Package found = packageService.findPackageById(searchId);
            if (found != null) {
                System.out.println(found);
            } else {
                System.out.println("Package not found.");
            }
            break;
        case 4:
            try {
                System.out.print("Enter Package Id to calculate cost: ");
                String cid = sc.nextLine();
                packageService.calculatePackageCost(cid);
                Package updated = packageService.findPackageById(cid);
                System.out.println("Cost calculated.");
                System.out.printf("Package Cost for %s is: %.2f\n", updated.getPackageId(),
updated.getPackageCost());
            } catch (InvalidPackageIdException e) {
                System.out.println(e.getMessage());
            }
            break;
        case 5:
            System.out.println("Exiting program.");
            break;
        default:
            System.out.println("Invalid choice.");
      }
    } while (choice != 5);
    sc.close();
```

```
  }
}
```

PackageDao.java

```
package com.dao;
import java.util.List;
import com.model.Package;
public interface PackageDao {
  void addPackage(Package p);
  List<Package> fetchAllPackages();
  Package findPackageById(String packageId);
  void calculatePackageCost(Package p);
}
```

PackageDaoImpl.java

```
package com.dao;
import java.util.ArrayList;
import java.util.List;
import com.model.Package;
public class PackageDaoImpl implements PackageDao {
  List<Package> packageList = new ArrayList<>();
  @Override
  public void addPackage(Package p) {
    packageList.add(p);
  }
  @Override
  public List<Package> fetchAllPackages() {
    return packageList;
  }
  @Override
  public Package findPackageById(String packageId) {
    for (Package p : packageList) {
      if (p.getPackageId().equals(packageId)) {
        return p;
      }
    }
    return null;
  }
  @Override
  public void calculatePackageCost(Package p) {
    double cost = p.getBasicFare() * p.getNoOfDays();
    double discount = 0;
    if (p.getNoOfDays() > 5 && p.getNoOfDays() <= 8) {
      discount = cost * 0.03;
```

```
    } else if (p.getNoOfDays() > 8 && p.getNoOfDays() <= 10) {
        discount = cost * 0.05;
    } else if (p.getNoOfDays() > 10) {
        discount = cost * 0.07;
    }
    double costAfterDiscount = cost - discount;
    double gst = costAfterDiscount * 0.12;
    double finalCost = costAfterDiscount + gst;
    p.setPackageCost(finalCost);
  }
}
```

InvalidPackageIdException.java

```
package com.exception;
public class InvalidPackageIdException extends Exception {
  public InvalidPackageIdException(String message) {
    super(message);
  }
}
```

Package.java

```
package com.model;
public class Package {
  private String packageId;
  private String sourcePlace;
  private String destinationPlace;
  private int noOfDays;
  private double basicFare;
  private double packageCost;
  public String getPackageId() {
    return packageId;
  }
  public void setPackageId(String packageId) {
    this.packageId = packageId;
  }
  public String getSourcePlace() {
    return sourcePlace;
  }
  public void setSourcePlace(String sourcePlace) {
    this.sourcePlace = sourcePlace;
  }
  public String getDestinationPlace() {
    return destinationPlace;
  }
```

```
  public void setDestinationPlace(String destinationPlace) {
     this.destinationPlace = destinationPlace;
  }
  public int getNoOfDays() {
     return noOfDays;
  }
  public void setNoOfDays(int noOfDays) {
     this.noOfDays = noOfDays;
  }
  public double getBasicFare() {
     return basicFare;
  }
  public void setBasicFare(double basicFare) {
     this.basicFare = basicFare;
  }
  public double getPackageCost() {
     return packageCost;
  }
  public void setPackageCost(double packageCost) {
     this.packageCost = packageCost;
  }
  @Override
  public String toString() {
     return String.format("Package [packageId=%s, sourcePlace=%s, destinationPlace=%s,
noOfDays=%d, basicFare=%.2f, packageCost=%.2f]",
           packageId, sourcePlace, destinationPlace, noOfDays, basicFare, packageCost);
  }
}
```

PackageService.java

```
package com.service;
import java.util.List;
import com.model.Package;
import com.exception.InvalidPackageIdException;
public interface PackageService {
  void addPackage(Package p) throws InvalidPackageIdException;
  List<Package> fetchAllPackages();
  Package findPackageById(String packageId);
  void calculatePackageCost(String packageId) throws InvalidPackageIdException;
}
```

PackageServiceImpl.java

```
package com.service;
import java.util.List;
```

```java
import com.dao.PackageDao;
import com.dao.PackageDaoImpl;
import com.model.Package;
import com.exception.InvalidPackageIdException;
public class PackageServiceImpl implements PackageService {
  PackageDao packageDao = new PackageDaoImpl();
  private boolean validatePackageId(String packageId) {
     return packageId.length() == 7;
  }
  @Override
  public void addPackage(Package p) throws InvalidPackageIdException {
     if (!validatePackageId(p.getPackageId())) {
        throw new InvalidPackageIdException("Invalid Package Id");
     }
     packageDao.addPackage(p);
  }
  @Override
  public List<Package> fetchAllPackages() {
     return packageDao.fetchAllPackages();
  }
  @Override
  public Package findPackageById(String packageId) {
     return packageDao.findPackageById(packageId);
  }
  @Override
  public void calculatePackageCost(String packageId) throws InvalidPackageIdException {
     Package p = packageDao.findPackageById(packageId);
     if (p == null) {
        throw new InvalidPackageIdException("Invalid Package Id");
     }
     packageDao.calculatePackageCost(p);
  }
}
```