Delivery Agency V1

AgencyController

```
package com.controller;

import java.util.List;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.exception.NoOrdersFoundException;
import com.model.Order;
import com.service.AgencyService;

//use appropriate annotation to configure AgencyController as Controller
@Controller
public class AgencyController {

	//Use appropriate annotation
	@Autowired
	private AgencyService service;

	@RequestMapping(value = "/searchPage", method = RequestMethod.GET)
	public String searchPage(@ModelAttribute("orderBean") Order orderBean) {
```

```
		return "searchPage";
	}



	//invoke the service class - searchDeliveryOrders method.
	@RequestMapping(value = "/orderList", method = RequestMethod.POST)
	public String searchDeliveryOrders(@ModelAttribute("orderBean") @Valid Order orderBean, BindingResult result,
			ModelMap model) throws NoOrdersFoundException  {
		//fill the code
		if (result.hasErrors()) {
			return "searchPage";
		} else {
			List<Order> ordersFound = service.searchDeliveryOrders(orderBean);
			if(ordersFound.isEmpty()) {
				throw new NoOrdersFoundException("No orders found for this search criteria");
			} else {
				model.addAttribute("orders", ordersFound);
				return "deliveryListPage";
			}

		}
	}

	@ExceptionHandler(NoOrdersFoundException.class)
	public ModelAndView exceptionHandler(Exception e) {
	   mav.addObject("exception", e);
	   mav.setViewName("exceptionPage");
	   return mav;
	}
}
```

DeliveryAgencyApplication

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
public class DeliveryAgencyApplication {

	public static void main(String[] args) {
		SpringApplication.run(DeliveryAgencyApplication.class, args);
	}

}
```

NoOrdersFoundException

```
package com.exception;

public class NoOrdersFoundException extends Exception {
	private static final long serialVersionUID = 1L;

	public NoOrdersFoundException(String msg) {
		super(msg);
	}

}
```

Order

```
package com.model;

import javax.validation.constraints.NotEmpty;

import org.springframework.stereotype.Component;

//pojo class with required attributes,getters and setters
@Component
public class Order {
	private String orderId;
	private String orderFrom;
	private String deliveryTo;
	// fill the code

	private String deliveryDate;

	private String deliveryStatus;
	public String getOrderId() {
		return orderId;
	}
	public void setOrderId(String orderId) {
		this.orderId = orderId;
	}
	public String getOrderFrom() {
		return orderFrom;
	}
	public void setOrderFrom(String orderFrom) {
		this.orderFrom = orderFrom;
	}
	public String getDeliveryTo() {
		return deliveryTo;
```

```
	}
	public void setDeliveryTo(String deliveryTo) {
		this.deliveryTo = deliveryTo;
	}
	public String getDeliveryDate() {
		return deliveryDate;
	}
	public void setDeliveryDate(String deliveryDate) {
		this.deliveryDate = deliveryDate;
	}

	public String getDeliveryStatus() {
		return deliveryStatus;
	}
	public void setDeliveryStatus(String deliveryStatus) {
		this.deliveryStatus = deliveryStatus;
	}
	public Order(String orderId, String orderFrom, String deliveryTo, String deliveryDate, String
deliveryStatus) {
		super();
		this.orderId = orderId;
		this.orderFrom = orderFrom;
		this.deliveryTo = deliveryTo;
		this.deliveryDate = deliveryDate;
		this.deliveryStatus = deliveryStatus;
	}

	public Order()	{}

}
```

AgencyService

```
package com.service;


import java.util.ArrayList;


import org.springframework.stereotype.Service;


import com.model.Order;


//use appropriate annotation to configure AgencyService as a Service
@Service
public class AgencyService {
	ArrayList<Order> list = new ArrayList<Order>();


	//search orders and return the ArrayList<Order>
	public ArrayList<Order> searchDeliveryOrders(Order orderBean) {
		ArrayList<Order> foundOrders = new ArrayList<>();
		for(Order : list) {
			if(order.getDeliveryDate().equals(orderBean.getDeliveryDate()) &&
order.getDeliveryStatus().equals(orderBean.getDeliveryStatus())) {
				foundOrders.add(order);
			}
		}
		return foundOrders;
	}


	public AgencyService() {
		 addOrders();
	}


	public  void addOrders(){			// don't change this code
```

```
        list.add(new Order("DF1234233","AJIO trends","#412, Shipitha cascade, Ganapathy, PIN-641001","10/02/2021","pending"));

        list.add(new Order("FGF7788","Amazon india","#12, shardhalotus, Ram Nagar, PIN-641001", "02/02/2021","pending"));

        list.add(new Order("AF5546","Mynthra","#12, Anandha nilayam,Kuniamuthur, PIN-641001","08/02/2021","pending"));

        list.add(new Order("DF5543","Flip cart","#201, Shipitha cascade, Saravanampatti, PIN-641001","10/02/2021","delivered"));


        list.add(new Order("DX17788","AJIO trends","#34/45, RV Homes, 5th cross, Anssari street, PIN-600001","01/02/2021","pending"));

        list.add(new Order("RF55346","Amazon india","#212/4, 10th cross, T-nagar, PIN-60008","03/02/2021","delivered"));

        list.add(new Order("DF88994","Mynthra","#556/4, 5th cross, Thambaram, PIN-60010","12/02/2021","pending"));

        list.add(new Order("FT51122","AJIO trends","#556/4,Nila paradise, Thambaram, PIN-60010","03/02/2021","delivered"));

        list.add(new Order("LK44338","Flip cart","#412/4, 10th cross, Velacheri, PIN-60009","03/02/2021","delivered"));


        list.add(new Order("JK99008","Amazon india","#45, 12Th main , GM Palya, PIN-560075","03/02/2021","pending"));

        list.add(new Order("MJ99765","Flip cart","#67/2, 3rd cross , Indra nagar, PIN-560077","08/02/2021","delivered"));

        list.add(new Order("NK66754","Mynthra","#302, Sakthi enclave ,1st main , Malleshpalya, PIN-560075","08/02/2021","delivered"));



    }
}
```

Digital Home Brokering

BrokerController

```
package com.controller;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.model.Home;
import com.service.BrokerService;

//use appropriate annotation to configure BrokerController as Controller
@Controller
public class BrokerController {

	//Use appropriate annotation above this property
	@Autowired
	private BrokerService service;

	@RequestMapping(value = "/searchPage", method = RequestMethod.GET)
	public String searchPage(@ModelAttribute("homeBean") Home homeBean)	{

		return "searchPage";
	}

	//invoke the service class - searchHome method.
```

```
	public String searchHome(@ModelAttribute("homeBean") @Valid Home homeBean,
BindingResult result,
				ModelMap model) {
		if (result.hasErrors()) {
			return "searchPage";
		}
		model.addAttribute("homes", service.searchHome(homeBean));
		return "homeListPage";
	}
}
```

DigitalHomeBrokerApplication

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@ComponentScan({"com.*"})
public class DigitalHomeBrokerApplication {

	public static void main(String[] args) {
		SpringApplication.run(DigitalHomeBrokerApplication.class, args);
	}

}
```

Home

```
package com.model;
```

```
import javax.validation.constraints.Min;

import org.springframework.stereotype.Component;

//pojo class with required attributes,getters and setters
@Component
public class Home {

	private String location;
	private String address;
	private String contactNumber;
	private String contactPerson;
	@Min(value = 5000, message = "Minimum rent should be 5000")
	private double rent;
	public double getRent() {
		return rent;
	}
	public void setRent(double rent) {
		this.rent = rent;
	}
	public String getLocation() {
		return location;
	}
	public void setLocation(String location) {
		this.location = location;
	}
	public String getAddress() {
		return address;
	}
	public void setAddress(String address) {
		this.address = address;
```

```
}
public String getContactNumber() {
	return contactNumber;
}
public void setContactNumber(String contactNumber) {
	this.contactNumber = contactNumber;
}
public String getBhkType() {
	return bhkType;
}
public void setBhkType(String bhkType) {
	this.bhkType = bhkType;
}
public String getContactPerson() {
	return contactPerson;
}
public void setContactPerson(String contactPerson) {
	this.contactPerson = contactPerson;
}

public Home(String location, String address, String contactNumber, String bhkType) {
	super();
	this.location = location;
	this.address = address;
	this.contactNumber = contactNumber;
	this.bhkType = bhkType;
	this.contactPerson = contactPerson;
	this.rent=rent;
}
public Home()
{}
```

```
}
```

BrokerService

```
package com.service;

import java.util.ArrayList;

import org.springframework.stereotype.Service;

import com.model.Home;

//use appropriate annotation to configure BrokerService as a Service
@Service
public class BrokerService {

	ArrayList<Home> list = new ArrayList<Home>();

	// search homes/apartments from the list and return it as a ArrayList
	public ArrayList<Home> searchHome(Home homeBean) {
		ArrayList<Home> resultList = new ArrayList<Home>();
		for (Home : list) {
			if (home.getLocation().equalsIgnoreCase(homeBean.getLocation())
					&&
home.getBhkType().equalsIgnoreCase(homeBean.getBhkType())
					&& home.getRent() == homeBean.setRent()) {
				resultList.add(home);
			}
		}
		return resultList;
```

```
	}


	public BrokerService() {
		addHomes();
	}


	// Dont modify this code
	public ArrayList<Home> addHomes() {
		list.add(new Home("Coimbatore", "#412, Shipitha cascade, Ganapathy, PIN-641001", "9917678456", "1BHK",
				"Malavika", 8000));
		list.add(new Home("Coimbatore", "#12, shardhalotus, Ram Nagar, PIN-641001", "9917100989", "2BHK", "Karthik",
				12000));
		list.add(new Home("Coimbatore", "#12, Anandha nilayam,Kuniamuthur, PIN-641001", "9917100989", "2BHK", "Karthik",
				12000));
		list.add(new Home("Coimbatore", "#201, Shipitha cascade, Saravanampatti, PIN-641001", "9917123456", "3BHK",
				"Sowparnika", 17000));


		list.add(new Home("Chennai", "#34/45, RV Homes, 5th cross, Anssari street, PIN-600001", "7788900989", "1BHK",
				"Mukunthan", 6000));
		list.add(new Home("Chennai", "#212/4, 10th cross, T-nagar, PIN-60008", "9917100989", "2BHK", "Annesh", 14000));
		list.add(new Home("Chennai", "#556/4, 5th cross, Thambaram, PIN-60010", "9917100989", "2BHK", "Jisha", 14000));
		list.add(new Home("Chennai", "#556/4,Nila paradise, Thambaram, PIN-60010", "9917100989", "2BHK", "Jisha",
				14000));
		list.add(
				new Home("Chennai", "#412/4, 10th cross, Velacheri, PIN-60009", "9917156989", "3BHK", "Rakesh", 18000));
```

```
		list.add(
				new Home("Bangalore", "#45, 12Th main , GM Palya, PIN-560075",
"9917122119", "1BHK", "Senthil", 11000));
		list.add(new Home("Bangalore", "#67/2, 3rd cross , Indra nagar, PIN-560077",
"9944550989", "2BHK", "RamKumar",
				17000));
		list.add(new Home("Bangalore", "#302, Sakthi enclave ,1st main , Malleshpalya, PIN-
560075", "9917668889",
				"3BHK", "Saradha", 25000));
		return list;

	}
}
```

FarmFreshOnlineStore

Shopping controller

```
package com.controller;


import java.util.ArrayList;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;

import com.exception.NoStockException;
import com.model.Product;
import com.service.ShoppingService;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.ModelAttribute;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

//use appropriate annotation to configure ShoppingController as Controller
@Controller
public class ShoppingController {

	@Autowired
	private ShoppingService;

	@RequestMapping(value = "/showPage , method"= RequestMethod.GET)
	public String showPage(@ModelAttribute("product")Product product) {

		return "showPage";
	}

	@ModelAttribute("productNames")
	public  List<String> populateProduct(){
		List<String> productNames = new ArrayList<String>();
			productNames.add("Apple");
			productNames.add("Orange");
			productNames.add("Grapes");
			productNames.add("Mango");
			productNames.add("JackFruit");
			return productNames;
	}

	//invoke the service class - calculateCostAndUpdate method.
	@RequestMapping(value = "/calculate", method = RequestMethod.POST)
```

```
	public String calculateCost(@ModelAttribute("product")@Valid Product, BindingResult result,ModelMap model) throws  NoStockException {

		if(result.hasErrors()){

			return ("showPage");
		}
		else{
			double cost=shoppingService.calculateCost(product);
			model.addAttribute("costPerKg",product.getCostPerKg());
			model.addAttribute("name",product.getProductName());
			model.addAttribute("quantity",product.getQuantity());

		}

		return ("billDesk");
	}


	@ExceptionHandler(NoStockException. class)
	public ModelAndView exceptionHandler(Exception e) {

		ModelAndView mav = new ModelAndView();
		mav.addObject("message", e.getMessage());
		mav.setViewName("exceptionPage");
		return mav;
	}

}
```

FarmFreshApplication

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@ComponentScan({"com.*"})
public class FarmFreshApplication {

	public static void main(String[] args) {
		SpringApplication.run(FarmFreshApplication.class, args);
	}

}
```

NoStockexception

```
package com.exception;

public class NoStockException extends Exception{
	private static final long serialVersionUID = 1L;
public NoStockException(String msg) {

	super(msg);
}

}
```

Product

```
package com.model;
```

```
import javax.validation.constraints.Min;
import javax.validation.constraints.NotEmpty;

import org.springframework.stereotype.Component;

@Component
public class Product {
	private String productName ;
	@Min(value=1,message="Minimum quantity should be 1Kg")
	private int quantity;


public String getProductName() {
		return productName;
	}

	public void setProductName(String productName) {
		this.productName = productName;
	}


	public int getQuantity() {
		return quantity;
	}


	public void setQuantity(int quantity) {
		this.quantity = quantity;
	}
```

```
	public double getCostPerKg() {
		return costPerKg;
	}


	public void setCostPerKg(double costPerKg) {
		this.costPerKg = costPerKg;
	}


public Product() {}
}
```

Shopping Service

```
package com.service;

import java.util.HashMap;
import java.util.Map;
import java.util.Map.Entry;

import org.springframework.stereotype.Service;

import com.exception.NoStockException;
//use appropriate annotation to configure ElectricityService as a Service
@Service
public class ShoppingService {

	public double calculateCost(Product product) throws  NoStockException {
	double totalCost=0.0;
```

```
	Map<String,Integer> stock=getProductStock();
	for (Entry<String, Integer> entry : stock.entrySet()) {
		if(product.getProductName().equalsIgnoreCase(entry.getKey())) {

			if(product.getQuantity()>entry.getValue()) {
				throw new NoStockException("No enough stock for product
"+product.setProductName());
			}
		}
	}

	if(product.getProductName().equals("Apple")) {
		product.setCostPerKg(250);
	}
	else if(product.getProductName().equals("JackFruit")) {
		product.setCostPerKg(75);
	}
	else if(product.getProductName().equals("Orange")) {
		product.setCostPerKg(90);
	}
	else if(product.getProductName().equals("Mango")) {
		product.setCostPerKg(60);
	}
	else if(product.getProductName().equals("Grapes")) {
		product.setCostPerKg(150);
	}

	totalCost=product.getQuantity()*product.getCostPerKg();

	return totalCost;
	}
```

```
	public  Map<String,Integer> getProductStock(){
		Map<String,Integer> stock = new HashMap<String,Integer>();
		stock.put("Apple",50);
		stock.put("Grapes",10);
		stock.put("Orange",30);
		stock.put("Mango",75);
		stock.put("JackFruit",25);
			return stock;
	}
}
```

Flawless Academy

AcademyController

```
package com.controller;

import java.util.HashMap;
import java.util.Map;
import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.model.Academy;
import com.service.AcademyService;
```

```
@Controller
public class AcademyController {

	@Autowired
	private AcademyService service;

	@RequestMapping(value = "/enrollmentPage", method = RequestMethod.GET)
	public String showPage(@ModelAttribute("academyBean") Academy academyBean) {
		System.out.println("in controller");
		return "enrollmentPage";
	}
	@ModelAttribute("programList")
	public  Map<String, String> buildState(){

		Map<String, String> progMap = new HashMap<String, String>();
		progMap.put("ClassicalDance", "ClassicalDance");
		progMap.put("KarnaticVocals", "KarnaticVocals");
		progMap.put("WesternDance","WesternDance");
		progMap.put("Drawing", "Drawing");
		progMap.put("Instruments","Instruments");

		return progMap;

	}

	@RequestMapping(value = "/progEstimation", method = RequestMethod.POST)
	public String calculateProgramCost(@ModelAttribute("academyBeanMethod") @Valid
Academy academyBean, BindingResult result,
			ModelMap model) {
		if (result.hasErrors()) {
			System.out.println("Error : " + result.toString());
```

```
                return "enrollmentPage";
        }
        double cost=service.calculateProgramCost(academyBean);
        model.addAttribute("cost", cost);
        return "estimationPage";
    }


}
```

FlawlessAcademyApplication

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@ComponentScan({"com.*"})
public class FlawlessAcademyApplication {

    public static void main(String[] args) {
        SpringApplication.run(FlawlessAcademyApplication.class, args);
    }

}
```

Academy

```
package com.model;
```

```
import javax.validation.constraints.Min;
import javax.validation.constraints.Max;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component
public class Academy {

	private String program;

	private double costPerSession;
	@Min(value = 2, message = "Minimum 2 Sessions/Week")
	@Max(value = 5, message = "Maximum 5 Sessions/Week")
	private int sessionsPerWeek;

	public int getSessionsPerWeek() {
		return sessionsPerWeek;
	}

	public void setSessionsPerWeek(int sessionsPerWeek) {
		this.sessionsPerWeek = sessionsPerWeek;
	}

	public Academy()
	{

	}

	public int getWeeksPerMonth() {
```

```
		return weeksPerMonth;
	}
	public void setWeeksPerMonth(int weeksPerMonth) {
		this.weeksPerMonth = weeksPerMonth;
	}
	public double getCostPerSession() {
		return costPerSession;
	}
	public void setCostPerSession(double costPerSession) {
		this.costPerSession = costPerSession;
	}

	public String getProgram() {
		return program;
	}
	public void setProgram(String program) {
		this.program = program;
	}

}
```

AcademyService

```
package com.service;

import org.springframework.stereotype.Service;

import com.model.Academy;

public class AcademyService {

	public double calculateProgramCost (Academy academyBean) {
```

```
		double cost=0.0;
		academyBean.setWeeksPerMonth(4);
		System.out.println(academyBean.getSessionsPerWeek()+"
"+academyBean.getProgram()+" "+academyBean.getWeeksPerMonth());
		if(academyBean.getProgram().equalsIgnoreCase("ClassicalDance") &&
academyBean.getSessionsPerWeek()>=1)
		{
			academyBean.setCostPerSession(150.0);

	cost=academyBean.getCostPerSession()*academyBean.getSessionsPerWeek()*academyBea
n.getWeeksPerMonth();
		}
	else if(academyBean.getProgram().equalsIgnoreCase("KarnaticVocals") &&
academyBean.getSessionsPerWeek()>=1 )
	{
		academyBean.setCostPerSession(100.0);

	cost=academyBean.getCostPerSession()*academyBean.getSessionsPerWeek()*academyBea
n.getWeeksPerMonth();
	}
	else if(academyBean.getProgram().equalsIgnoreCase("WesternDance") &&
academyBean.getSessionsPerWeek()>=1)
	{
		academyBean.setCostPerSession(125.0);

	cost=academyBean.getCostPerSession()*academyBean.getSessionsPerWeek()*academyBea
n.getWeeksPerMonth();
	}
	if(academyBean.getProgram().equalsIgnoreCase("Drawing") &&
academyBean.getSessionsPerWeek()>=1)
	{
		academyBean.setCostPerSession(130.0);

	cost=academyBean.getCostPerSession()*academyBean.getSessionsPerWeek()*academyBea
n.getWeeksPerMonth();
```

```
	}
else if(academyBean.getProgram().equalsIgnoreCase("Instruments") &&
academyBean.getSessionsPerWeek()>=1)
{
	academyBean.setCostPerSession(200.0);
	cost=academyBean.getCostPerSession()*academyBean.getSessionsPerWeek()*academyBea
n.getWeeksPerMonth();
}
	return cost;

	}

}
```

Snap Fitness Centre

Fitness Controller

```
package com.controller;

import java.util.HashMap;
import java.util.Map;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;

import javax.validation.Valid;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

import com.model.FitnessCenter;
```

```
import com.service.FitnessService;

//use appropriate annotation to configure SnapFitnessController as Controller
@Controller
public class FitnessController {

//	Use appropriate annotation
   @Autowired
	private FitnessService service;

	@ModelAttribute("packageList")
	 public Map<String, String> buildState() {
		Map<String, String> pairs = new HashMap();
		pairs.put("Aerobics","Aerobics");
		pairs.put("Zumba","Zumba");
		pairs.put("PersonalTraining","PersonalTraining");
		pairs.put("Yoga","Yoga");
		pairs.put("RegularGYM","RegularGYM");
		return pairs;
	}
	@GetMapping("/snapEnquiryPage")
	 public String snapEnquiryPage( @ModelAttribute("snap") FitnessCenter fitnessCenter){
		 return "snapEnquiryPage";
	 }
   @PostMapping("/<packEstimation>")
	public String calculatePackageCost(@Valid @ModelAttribute("snap")FitnessCenter fitness,
BindingResult result,
				ModelMap model) {
	if(result.hasErrors()){
	     model.addAttribute("messege", "Number of months should be minimum one");
				return "snapEnquiryPage";
```

```
		}
		model.addAttribute("cost",service.PackageCost(fitness));
		return "snapEstimation";
	}


}
```

SnapFitnessApplication

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication
@ComponentScan({"com.*. context.annotation "})
public class SnapFitnessApplication {

	public static void main(String[] args) {
		SpringApplication.run(SnapFitnessApplication.class, args);
	}

}
```

FitnessCentre

```
package com.model;
import javax.validation.constraints.Min;
import org.springframework.stereotype.Component;
```

```
//pojo class with required attributes,getters and setters

//use appropriate annotation to make this class as component class
@Component
@Aspect
public class FitnessCenter {

	private String preferredPackage;
	private double costPerMonth;
	// use appropriate annotation for validating numberOfMonths attribute
	@Min(value= 1,message="Number of months should be minimum one")
	private int numberOfMonths;



	public String getPreferredPackage() {
		return preferredPackage;
	}
	public void setPreferredPackage(String preferredPackage) {
		this.preferredPackage = preferredPackage;
	}
	public double getCostPerMonth() {
		return costPerMonth;
	}
	public void setCostPerMonth(double costPerMonth) {
		this.costPerMonth = costPerMonth;
	}
	public int getNumberOfMonths() {
		return numberOfMonths;
	}
	public void setNumberOfMonths(int numberOfMonths) {
```

```
		this.numberOfMonths = numberOfMonths;
	}


}
```

FitnessService

```
package com.service;

import com.model.FitnessCenter;

import org.springframework.stereotype.Service;

//use appropriate annotation to configure SnapFitnessService as a Service class
@Service
public class FitnessService {

	//calculate the Package cost and return the rent amount
	public double calculatePackageCost (FitnessCenter fitness) {

		float packageCost=0.0;
		// fill the code
		String serviceName = fitness.getPreferredPackage();
		int months = fitness.getNumberOfMonths();
		if(serviceName.equals("Aerobics"))
		{
			packageCost = months*1700.0;
		}
		else if(serviceName.equals("Zumba"))
		{
			packageCost = months*1750.0;
		}
```

```
		else if(serviceName.equals("RegularGYM"))
		{
			packageCost = months*1280.0;
		}
		else if(serviceName.equals("PersonalTraining"))
		{
			packageCost = months*2500.0;
		}
		else if(serviceName.equals("Yoga"))
		{
			packageCost = months*1400.0;
		}

		return packageCost;


	}

}
```