Assignment 3

Connor McIntire 917610885

Mobile Development

CS3310

Fall 2024

Meal Combo Class

Prompt: OK lets do my meal combo class

Code Generated:

```
package com.example.demo; // Adjust this to match your actual package

import java.util.List;

public class MealCombo {
    private Long id; // Unique identifier for the combo
    private String mealType; // Type of meal (breakfast, lunch, dinner)
    private List<String> comboItems; // List of items in the combo

    // Default constructor
    public MealCombo() {}

    // Parameterized constructor
    public MealCombo(Long id, String mealType, List<String> comboItems) {
        this.id = id;
        this.mealType = mealType;
        this.comboItems = comboItems;
    }

    // Getter and setter for id
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    // Getter and setter for mealType
    public String getMealType() {
        return mealType;
    }

    public void setMealType(String mealType) {
        this.mealType = mealType;
    }

    // Getter and setter for comboItems
    public List<String> getComboItems() {
        return comboItems;
    }

    public void setComboItems(List<String> comboItems) {
        this.comboItems = comboItems;
    }
}
```

Help Request Class

Prompt: Now I would like the other class called HelpRequest I have provided you the outline of my instructions before this

Code Generated:

```
package com.example.demo; // Adjust this to match your actual package

import java.time.LocalDateTime;

public class HelpRequest {
    private Long id; // Unique identifier for the help request
    private String requesterName; // Name of the person requesting help
    private LocalDateTime timestamp; // When the help was requested
    private String helpType; // Type of help requested (medical, feeding, other)
    // Default constructor
    public HelpRequest() {}

    // Parameterized constructor
    public HelpRequest(Long id, String requesterName, LocalDateTime timestamp, String helpType) {
        this.id = id;
        this.requesterName = requesterName;
        this.timestamp = timestamp;
        this.helpType = helpType;
    }

    // Getter and setter for id
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    // Getter and setter for requesterName
    public String getRequesterName() {
        return requesterName;
    }

    public void setRequesterName(String requesterName) {
        this.requesterName = requesterName;
    }

    // Getter and setter for timestamp
    public LocalDateTime getTimestamp() {
        return timestamp;
    }

    public void setTimestamp(LocalDateTime timestamp) {
        this.timestamp = timestamp;
    }

    // Getter and setter for helpType
    public String getHelpType() {
        return helpType;
    }
    public void setHelpType(String helpType) {
        this.helpType = helpType;
    }
}
```

Meal Combo and Help Request Interfaces

Prompt: ok what should I do next?

Code Generated:

```
package com.example.demo.repository;

import com.example.demo.MealCombo;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface MealComboRepository extends JpaRepository<MealCombo, Long> {
    // Custom query methods (if needed) can be defined here
}

package com.example.demo.repository;

import com.example.demo.HelpRequest;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface HelpRequestRepository extends JpaRepository<HelpRequest, Long> {
    // Custom query methods (if needed) can be defined here
}
```

Meal Combo and Help Request Controllers

Prompt: Next I would like to do the controllers for my project. Please provide me guidance on what to do and explain each line of code.

Code Generated:

```
package com.example.demo.controller;
```

```
import com.example.demo.MealCombo;
import com.example.demo.repository.MealComboRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/v1/meal")
public class MealComboController {

   @Autowired
   private MealComboRepository mealComboRepository;

   @GetMapping("/{mealType}")
   public List<MealCombo> getMealCombos(@PathVariable String mealType) {
      // Retrieve meal combos based on mealType (implement filtering logic)
      return mealComboRepository.findAll(); // Placeholder
   }

   @GetMapping("/{mealType}/{id}")
   public ResponseEntity<MealCombo> getMealComboById(@PathVariable String mealType,
@PathVariable Long id) {
      return mealComboRepository.findById(id)
            .map(ResponseEntity::ok)
            .orElse(ResponseEntity.notFound().build());
   }

   @PostMapping("/{mealType}/{id}")
   public ResponseEntity<String> orderMealCombo(@PathVariable String mealType,
@PathVariable Long id) {
      // Logic to place a meal order
      return ResponseEntity.ok("Order placed for meal combo " + id);
   }
}

package com.example.demo.controller;

import com.example.demo.HelpRequest;
import com.example.demo.repository.HelpRequestRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
```

```
@RestController
@RequestMapping("/api/v1/help")
public class HelpRequestController {

    @Autowired
    private HelpRequestRepository helpRequestRepository;

    @PostMapping
    public ResponseEntity<HelpRequest> createHelpRequest(@RequestBody HelpRequest helpRequest) {
        HelpRequest savedRequest = helpRequestRepository.save(helpRequest);
        return ResponseEntity.ok(savedRequest);
    }
}
```

Meal Combo Service Class

Prompt: Now I would like to set the available combos for each meal

Code Generated:

```
package com.example.demo;

import org.springframework.stereotype.Service;
import java.util.Arrays;
import java.util.List;

@Service
public class MealComboService {

    // Predefined breakfast combos
    private final List<MealCombo> breakfastCombos = Arrays.asList(
        new MealCombo(1L, MealCombo.MealType.BREAKFAST, Arrays.asList("sausage", "grit", "tea", "toast")),
        new MealCombo(2L, MealCombo.MealType.BREAKFAST, Arrays.asList("sausage", "toast", "coffee", "scrambled eggs")),
        new MealCombo(3L, MealCombo.MealType.BREAKFAST, Arrays.asList("sunny side up eggs", "toast", "water"))
    );

    // Predefined lunch combos
    private final List<MealCombo> lunchCombos = Arrays.asList(
        new MealCombo(4L, MealCombo.MealType.LUNCH, Arrays.asList("chicken salad", "bread roll", "iced tea")),
        new MealCombo(5L, MealCombo.MealType.LUNCH, Arrays.asList("turkey sandwich", "chips", "soda")),
        new MealCombo(6L, MealCombo.MealType.LUNCH, Arrays.asList("pasta", "garlic bread", "water"))
    );

    // Predefined dinner combos
    private final List<MealCombo> dinnerCombos = Arrays.asList(
        new MealCombo(7L, MealCombo.MealType.DINNER, Arrays.asList("grilled salmon", "mashed potatoes", "green beans")),
        new MealCombo(8L, MealCombo.MealType.DINNER, Arrays.asList("steak", "fries", "coke")),
        new MealCombo(9L, MealCombo.MealType.DINNER, Arrays.asList("vegetable stir-fry", "rice", "herbal tea"))
    );

    // Method to get combos based on meal type
    public List<MealCombo> getCombosByMealType(MealCombo.MealType mealType) {
        switch (mealType) {
            case BREAKFAST:
                return breakfastCombos;
            case LUNCH:
                return lunchCombos;
            case DINNER:
                return dinnerCombos;
            default:
                throw new IllegalArgumentException("Unknown meal type: " + mealType);
        }
    }
}
```

Help Request Service Class

Prompt: Now I would like you to do the same thing but for the help request service class

Code Generated:

```
package com.example.demo.service;
import org.springframework.stereotype.Service;
import com.example.demo.model.HelpRequest;
@Service
public class HelpRequestService {
    // Method to log a help request
    public void logHelpRequest(HelpRequest helpRequest) {
        // Logic to handle the help request
        // E.g., save to a database, send an email, etc.
        System.out.println("Help request received: " + helpRequest.getRequestDetails());
    }
}
```

Meal Order Classes

Prompt: Now I would like you to do the same thing but for meal orders

Code Generated:

```
package com.example.demo.model;

import javax.persistence.*;

@Entity
public class MealOrder {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String mealName;
    private int quantity;
    private String customerName;

    // Getters and Setters
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getMealName() {
        return mealName;
    }

    public void setMealName(String mealName) {
        this.mealName = mealName;
    }

    public int getQuantity() {
        return quantity;
    }

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

    public String getCustomerName() {
        return customerName;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }
}
package com.example.demo.repository;

import com.example.demo.model.MealOrder;
import org.springframework.data.jpa.repository.JpaRepository;

public interface MealOrderRepository extends JpaRepository<MealOrder, Long> {
}
package com.example.demo.controller;

import com.example.demo.model.MealOrder;
import com.example.demo.repository.MealOrderRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/mealorders")
public class MealOrderController {
    @Autowired
    private MealOrderRepository mealOrderRepository;

    // Place a meal order
    @PostMapping
    public ResponseEntity<MealOrder> placeMealOrder(@RequestBody MealOrder
mealOrder) {
        MealOrder savedOrder = mealOrderRepository.save(mealOrder);
        return new ResponseEntity<>(savedOrder, HttpStatus.CREATED);
    }

    // Get all meal orders
    @GetMapping
    public ResponseEntity<List<MealOrder>> getAllMealOrders() {
        List<MealOrder> orders = mealOrderRepository.findAll();
        return new ResponseEntity<>(orders, HttpStatus.OK);
    }

    // Cancel a meal order (delete)
    @DeleteMapping("/{id}")
    public ResponseEntity<Void> cancelMealOrder(@PathVariable Long id) {
        if (mealOrderRepository.existsById(id)) {
            mealOrderRepository.deleteById(id);
            return new ResponseEntity<>(HttpStatus.NO_CONTENT);
        } else {
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
        }
    }
}
```