

CCT College Dublin

Assignment Cover Page

Module Title: Cloud Based Web Applications

Assignment Title: Farm Manager

Lecturer Name: David Gonzalez

Student Name: Vitoria Régia Fernandes de Oliveira

Student No.: 2019153

Assignment Due Date: 17th May 2020 23:59

Summary

Introduction	3
Specific Requirements	4
Controller Class	5
Inputs for Post Method	10

Introduction

We were asked to develop a system to control the livestock of a farm. The system will allow us to do some calculations around the benefits of the farm as well as get a summary of the current stocking level of the same. The system will also need to be documented (word document or similar) up to a point that an engineer on your similar skill level, would be able to take over and develop new features.

Specific Requirements

The system will allow us to add new animals to the farm stock. The farm will only contain three types of animals:

- Cows
- Pigs
- Chickens

Each animal has 2 attributes:

- Type
- Weight

Each animal has a market value:

- Cows - €500
- Pigs - €250
- Chickens - €5

The animals can only be sold if they reach a certain weight:

- Cows - 300 KG
- Pigs 100 KG
- Chickens - 0.5 KG

With the following endpoints:

- Add a new animal.
- Calculate the average weight of each type of animal (one endpoint is sufficient, no need to build one per type of animal).
- How many animals of each type can be sold (weight requirements above) right now.
- What is the current value of the full farm stock: That is, the price of all the animals that can be sold right now.
- What is the current value of the farm assuming the price of each animal is set by a parameter in the HTTP request.

This is an example: - <http://localhost:8080/currentValue?cow=350&pig=120&chicken=1>

Controller class:

CAFarmController:

```
package ie.cct.CBWA.CAFarmProject;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.RequestBody;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
```

```
public class CAFarmController {
```

```
    private List<Farm> farmitem;
```

```
    public CAFarmController() {
```

```
        farmitem = new ArrayList<Farm>(); // farm items array list
```

```
    }
```

```
    // Post http://localhost:8080/add-animal
```

```
    // this method will add a new animal ( list in txt added to the folder with few
```

```
    // animals data)
```

```
    @PostMapping("add-animal")
```

```
    public SuccessResponse addAnimal(@RequestBody Farm animal) {
```

```
        farmitem.add(animal);
```

```
        int count = 0;
```

```
        count = farmitem.size();
```

```
        return new SuccessResponse(
```

```
            "Farm animal:" + animal.getAnimal() + " added. Now there are " + count + " animals in farm");
```

```
    }
```

```
    // Get http://localhost:8080/average-price
```

```
    // this method will calculate the average weight of added animals
```

```
    @GetMapping("average-weight")
```

```
    public Float avgWeight() {
```

```

        if (farmitem.size() == 0) {
            throw new NotFoundException(" No animal found in the Farm");
        }
        float weight = 0.0f;
        for (Farm animal : farmitem) {
            weight += animal.getWeight();
        }
        weight = weight / farmitem.size();
        return weight;
    }

    // Get http://localhost:8080/chickentosell

    // this method will check how many cows are available to be sold according to
    // their weight
    @GetMapping("chickentosell")
    public int chickentosell() {
        int c = 0;
        if (farmitem.size() == 0) {
            throw new NotFoundException(" No animal found in the Farm");
        }
        for (Farm animal : farmitem) {
            if ((animal.getWeight() >= 0.5) && (animal.getWeight() < 10)) {
                c++;
            }
        }
        return c;
    }

    // Get http://localhost:8080/cowtosell

    // this method will check how many cows are available to be sold according to
    // their weight

```

```

@GetMapping("cowtosell")
public int cowtosell() {
    int c = 0;

    if (farmitem.size() == 0) {
        throw new NotFoundException(" No animal found in the Farm");
    }
    for (Farm animal : farmitem) {
        if (animal.getWeight() >= 350) {

            c++;
        }
    }

    return c;
}

// Get http://localhost:8080/currentValue
// This method show what is the current value of the farm assuming the price of
// each animal is set by a parameter in the HTTP request. This is an example: -
// http://localhost:8080/currentValue?cow=350&pig=120&chicken=1

@GetMapping("currentValue")
public int currentValue(@RequestBody(required = true) int cow, @RequestBody(required = true) int pig, @RequestBody(required = true) int chicken) {
    int c = 0;
    int c2 = 0;
    int p = 0;
    int total = 0;

    if (farmitem.size() == 0) {
        throw new NotFoundException(" No animal found in the Farm");
    }
    for (Farm animal : farmitem) {
        if ((animal.getWeight() >= 0.5) && (animal.getWeight() < 10)) {

```

```

c++;

        } else if ((animal.getWeight() >= 100) && (animal.getWeight() < 200)) {
            p++;
        }
        else if (animal.getWeight() >= 350) {
            c2++;
        }

        int valuepig = 0;

int valuecow = 0;
int valuechicken = 0;
valuepig = p * pig;
valuecow = c2 * cow;
valuechicken = c * chicken;
total = valuepig + valuechicken + valuecow;
    }

        return total;
    }

    // Get http://localhost:8080/farmstock

    // this method will check how many animals are available to be sold according to
    // their weight, and will show the price of all the animals that can be sold
    // right now.

    @GetMapping("farmstock")
    public float farmstock() {
        int c = 0;
        int c2 = 0;
        int p = 0;
        int total = 0;
        if (farmitem.size() == 0) {
            throw new NotFoundException(" No animal found in the Farm");
        }
    }

```



```

for (Farm animal : farmitem) {
    if ((animal.getWeight() >= 0.5) && (animal.getWeight() < 10)) {
        c += animal.getPrice();

    } else if ((animal.getWeight() >= 100) && (animal.getWeight() < 200)) {
        p += animal.getPrice();
    }
    else if (animal.getWeight() >= 350) {
        c2 += animal.getPrice();
    }

    total = c + c2 + p;
}

return total;
}

// Get http://localhost:8080/pigtosell
// this method will check how many pigs are available to be sold according to their weight
@GetMapping("pigtosell")
public int pigtosell() {
    int p = 0;
    if (farmitem.size() == 0) {
        throw new NotFoundException(" No animal found in the Farm");
    }
    for (Farm animal : farmitem) {
        if ((animal.getWeight() >= 100) && (animal.getWeight() < 200)) {
            p++;
        }
    }

    return p;
}
}

```

Inputs for Post Method

For Post Method, feel free to utilise few of the inputs bellow:

```
{  
  "animal": "cow",  
  "price":500,  
  "weight": 350
```

```
}  
{  
  "animal": "pig",  
  "price":250,  
  "weight": 120
```

```
}  
{  
  "animal": "chicken",  
  "price":5,  
  "weight": 1
```

```
}  
{  
  "animal": "cow",  
  "price":500,  
  "weight": 310
```

```
}  
{  
  "animal": "pig",  
  "price":250,  
  "weight": 99
```

```
}  
{  
  "animal": "chicken",  
  "price":5,  
  "weight": 0.3
```

```
}  
{  
  "animal": "cow",  
  "price":500,  
  "weight": 315
```

```
}  
{  
  "animal": "pig",  
  "price":250,  
  "weight": 150
```

```
}  
{  
  "animal": "chicken",  
  "price":5,  
  "weight": 0.7
```

```
}
```