Technical Report

SKM3002 Group 3
Optimising Cake Production

STUDENT'S NAME
Aidil 'Aqif bin Mohd Pidaus

Presentation video
link:https://drive.google.com/drive/folders/1vvbzx8qvnoUqMFIPjm16-kSnA_tDMnp6?usp=sharing

INTRODUCTION

Optimising resources is the objective for all businesses to gain maximum profit with minimal to no loss. This case study focuses on optimising cake production for a bakery by calculating the maximum number of cakes that can be produced by a bakery based on the available ingredients. These calculations consider the amount of ingredients required for each cake based on the recipe as well as the amount of ingredients that are available. With this solution, bakeries can carry out their production processes with minimal waste and improve their profitability.

PROJECT BACKGROUND

The bakery want to optimise the amount of cakes bake based on their available ingredients but at the same time aiming to have minimal lost in their usage of ingredients. As example, Bakery A have 20kg of flours and 20L of milks while other amount of available ingredient is just enough to bake 10 Cake A then how much will Cake B can be produce in this case? This question which lead to create a production planner for bakery by letting the user of this system to enter the amount of available ingredients with the usage of appropriate unit for large scale production, recipe for each cake can be modified to meet the bakery needs and calculation for production is made based on these.

METHODOLOGY AND CALCULATION STEPS

Let R_C and R_R as recipe for chocolate cake and red velvet cake respectively. Each recipe has their own specific ingredients with specific quantities. These values are hard-coded into the source code.

$R_C = \{ flour_C, sugar_C, salt_C, sprinkles_C, oil_C, butter_C, egg_C, vanilla_C, baking\ powder_C, icing\ sugar_C, milk_C \}$

$R_R = \{ flour_R, sugar_R, salt_R, sprinkles_R, oil_R, butter_R, egg_R, vanilla_R, baking\ powder_R, icing\ sugar_R, milk_R \}$

Let A as available quantity for each ingredient. These values are prompted to user hence user will enter the available quality into the program.

$R_A = \{ flour_A, sugar_A, salt_A, sprinkles_A, oil_A, butter_A, egg_A, vanilla_A, baking\ powder_A, icing\ sugar_A, milk_A \}$

For each ingredient based on the recipe, calculation for maximum number of cakes can be produced is based on this equations.

$$\max\ \_cakes_C = \min\left(\frac{flour_A}{flour_C}, \quad \frac{sugar_A}{sugar_C}, \ldots, \quad \frac{milk_A}{milk_C}\right)$$

$$\max\ \_cakes_R = \min\left(\frac{flour_A}{flour_R}, \quad \frac{sugar_A}{sugar_R}, \ldots, \quad \frac{milk_A}{milk_R}\right)$$

The system will display the calculated maximum number of chocolate cakes and red velvet cakes that can be produced based on the available ingredients inserted by users.

CODING AND COMMENTS

main.py

```
from unit import Unit
from ingredient import Ingredient
from recipe import Recipe
from recipe_manager import RecipeManager
from production_planner import ProductionPlanner

# Define the recipes for chocolate and red velvet cakes
recipe_manager = RecipeManager()
recipe_manager.add_recipe(Recipe("Chocolate Cake", {
    'flour': Ingredient('flour', 0.2, Unit.KILOGRAM),
    'sugar': Ingredient('sugar', 0.15, Unit.KILOGRAM),
    'salt': Ingredient('salt', 0.001, Unit.KILOGRAM),
    'sprinkles': Ingredient('sprinkles', 0.005, Unit.KILOGRAM),
    'oil': Ingredient('oil', 0.100, Unit.LITRE),
    'butter': Ingredient('butter', 0.100, Unit.KILOGRAM),
    'egg': Ingredient('egg', 3, Unit.UNIT),
    'vanilla': Ingredient('vanilla', 0.001, Unit.LITRE),
    'baking powder': Ingredient('baking powder', 0.002, Unit.KILOGRAM),
    'icing sugar': Ingredient('icing sugar', 0.010, Unit.KILOGRAM),
    'milk': Ingredient('milk', 0.2, Unit.LITRE)
}))
recipe_manager.add_recipe(Recipe("Red Velvet Cake", {
    'flour': Ingredient('flour', 0.18, Unit.KILOGRAM),
    'sugar': Ingredient('sugar', 0.12, Unit.KILOGRAM),
    'salt': Ingredient('salt', 0.001, Unit.KILOGRAM),
    'sprinkles': Ingredient('sprinkles', 0.005, Unit.KILOGRAM),
    'oil': Ingredient('oil', 0.08, Unit.LITRE),
    'butter': Ingredient('butter', 0.08, Unit.KILOGRAM),
    'egg': Ingredient('egg', 2, Unit.UNIT),
    'vanilla': Ingredient('vanilla', 0.001, Unit.LITRE),
    'baking powder': Ingredient('baking powder', 0.002, Unit.KILOGRAM),
    'icing sugar': Ingredient('icing sugar', 0.010, Unit.KILOGRAM),
    'milk': Ingredient('milk', 0.2, Unit.LITRE)
}))

planner = ProductionPlanner(recipe_manager)
max_cakes = planner.calculate_production_plan()

for cake_name, max_cake in max_cakes.items():
    print(f"\nOptimum number of {cake_name}: {max_cake}")
```

production_planner.py

```
import math
from recipe_manager import RecipeManager

# Calculates the production plan based on available ingredients and recipes
class ProductionPlanner:
    def __init__(self, recipe_manager):
        self.recipe_manager = recipe_manager

    def calculate_production_plan(self):
        message = """Units for each Ingredient\n
        kilogram : flour, sugar, salt, sprinkles, butter, baking powder, icing sugar
        Litre : oil, vanilla, milk
        Units : egg\n
        """
        print(message)

        available_ingredients = {}
        for ingredient_name in self.recipe_manager.recipes['Chocolate
Cake'].ingredients.keys():
            amount = float(input(f"Enter the amount of {ingredient_name} available: "))
            available_ingredients[ingredient_name] = amount

        max_cakes = {}
        for cake_name in self.recipe_manager.recipes.keys():
            max_cake = float('inf')
            for ingredient_name, amount_unit in
self.recipe_manager.recipes[cake_name].ingredients.items():
                amount_required = amount_unit.amount
                if ingredient_name in available_ingredients:
                    amount_available = available_ingredients[ingredient_name]
                    max_cake = min(max_cake, amount_available // amount_required)
                else:
                    max_cake = 0
                    break

            # Round down the number of cakes to the nearest whole number
            max_cake = math.floor(max_cake)
            max_cakes[cake_name] = max_cake

        return max_cakes
```

recipe_manager.py

```python
from recipe import Recipe

# Manages recipes, allowing for addition and retrieval
class RecipeManager:
    def __init__(self):
        self.recipes = {}

    def add_recipe(self, recipe):
        self.recipes[recipe.name] = recipe

    def get_recipe(self, name):
        return self.recipes.get(name)
```

recipe.py

```python
from ingredient import Ingredient

# Represents a recipe with a name and a dictionary of ingredients
class Recipe:
    def __init__(self, name, ingredients):
        self.name = name
        self.ingredients = ingredients
```

ingredient.py

```python
from unit import Unit

# Represents an ingredient with name, amount, and unit
class Ingredient:
    def __init__(self, name, amount, unit):
        self.name = name
        self.amount = amount
        self.unit = unit
```

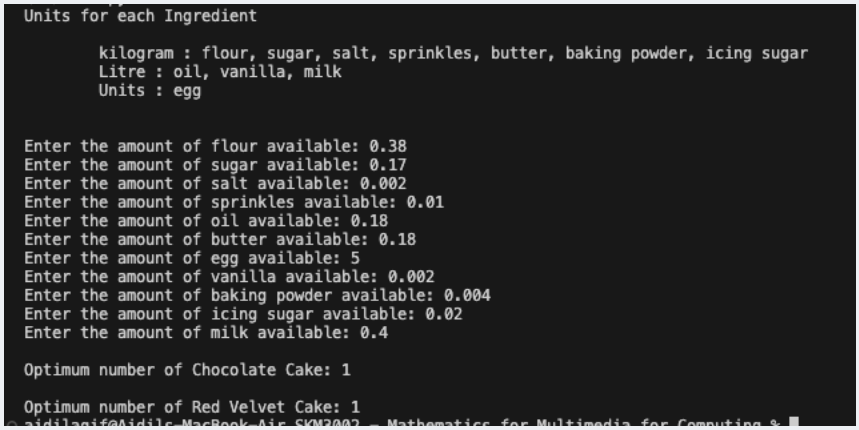
unit.py

```python
from enum import Enum

# Enum for different units of measurement
class Unit(Enum):
    KILOGRAM = 'kg'
    LITRE = 'L'
    UNIT = 'unit'
```

SAMPLE OUTPUT

expected output: 1 chocolate cake, 1 red velvet cake



expected output: 1 chocolate cake, 2 red velvet cake

CONCLUSION

Achieving the optimum cake production based on the ingredients that are available as well as the cakes recipe is essential aspects for a bakery to achieve maximum profits and minimizing waste. By implementing this system which calculates production plan based on available ingredients and recipes, the bakery can improve their production processes at the same time reducing costs.