# Technical Report

|  |  |
| --- | --- |
|  |  |
|  | SKM3002 Group 3Optimising Cake Production |
|  | STUDENT’S NAMEAidil ‘Aqif bin Mohd Pidaus |
| Presentation video link:https://drive.google.com/drive/folders/1vvbzx8qvnoUqMFIPjm16-kSnA\_tDMnp6?usp=sharing | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | INTRODUCTIONOptimising 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 BACKGROUNDThe 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 STEPSLet and 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.   = { , , , , , , , , , , }   = { , , , , , , , , , , }  Let as available quantity for each ingredient. These values are prompted to user hence user will enter the available quality into the program.   = { , , , , , , , , , , }  For each ingredient based on the recipe, calculation for maximum number of cakes can be produced is based on this equations. 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 COMMENTSmain.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 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 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 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 from enum import Enum  # Enum for different units of measurement  class Unit(Enum):  KILOGRAM = 'kg'  LITRE = 'L'  UNIT = 'unit' |
|  |

|  |  |
| --- | --- |
|  | SAMPLE OUTPUTexpected output: 1 chocolate cake, 1 red velvet cake    expected output: 1 chocolate cake, 2 red velvet cake |
|  |

|  |  |  |
| --- | --- | --- |
|  | CONCLUSIONAchieving 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. |  |
|  |  |