

PART A:

Problem Statement:

My client uses an organisational system combining paper and Microsoft excel, as such, she is not able to make various calculations automatically or quickly which results in inefficient use of her time.

Description of Scenario:

My client is Anamaria Robles, a local baker that sells baked goods to the International School Bangkok, as well as individuals in the community. When it comes to organisation, she initially writes her "... costs, ingredients and methods..." on paper and then inputs them into a Microsoft excel sheet. In doing so, she struggles with the organisation of her many recipes by different categories. Excel is not automatically able to sort by different parameters or calculate necessities such as "profit margins, costs per unit, costs per gram and ingredient ratios." . This forces my client to make calculations by hand, resulting in time inefficiency and possible errors.

Rationale for the Proposed Product:

A specialized computer GUI program, coded using Java and the NetBeans IDE, would be an optimal solution to the problem because of its ease-of-use and compatibility. A GUI program firstly can be made simplistic and specialized towards my client's needs. According to my client, she needs a "...dedicated application that suits [her] needs and only those needs". In comparison, the current system of excel contains various features that are useless to my client and thus only serve as confusing clutter. Java allows me to program the specific equations used by my client and simplify them to a press of a button. Furthermore, implementing sorting methods would allow my client to reorganise her recipes under different parameters by pressing a single button.

(Word Count: 268)

Success Criteria:

- Display a database of all my clients 12+ recipes
 - Include within the display and the database information on calories, cost, time, ingredients and method
- Allow my client to add recipes (and their respective fields) to the database.
 - Include the saving of data locally (within my client's computer)
 - Allow client to remove recipes from the database
- Allow my client to perform basic calculations that will aid her business including:
 - Profit margins
 - Conversion of recipe method and ingredients according to servings desired
 - Cost per unit
- Allow my client to sort the database of recipes:
 - Alphabetically
 - By price
 - By profit margin
- Allow the compilation of a menu that can be downloaded locally and shared.

File Edit Help

Input Recipes View Recipes Calculations

Name:

Servings:

Ingredients

Method

Item 1
Item 2
Item 3
Item 4
Item 5

Quantity:

Cups mL Tablespoon

Grams Teaspoon

Add Remove Submit

File Edit Help

Input Recipes

View Recipes

Calculations

Title 1

Title 2

Title 3

Title 4

File Edit Help

Input Recipes

View Recipes

Calculations

Sample Recipe ▾

Cost:

Profit Margins:

Servings Converter ▾

Calculate

Time to Complete:

Sales Price:

Main GUI
-jTabbedPane1: JTabbed Pan -inputPanel: JPanel -viewPanel: JPanel -calcPanel: JPanel -nameLab: JLabel -servingLav: JLabel -ingredientsLab: JLabel -methodsLab: JLabel -nameField: JTextField -servingField: JTextField -methodScroll: ScrollPane -methodsArea: JTextArea -submitBut: Button -adding: Button -ingScroll: ScrollPane -ingList: List -ingScroll2: ScrollPane -compList: List -removeIng: Button -quantityField: Field -quantityLab: JLabel -cupsBut: RadioButton -gramsBut: RadioButton -mlBut: RadioButton -tablespoonBut: RadioButton -teaspoonBut: RadioButton -viewScroll: ScrollPane -viewTable: Table -recipeBox: ComboBox -costLab: JLabel -profitLab: JLabel -servingsConvert: ComboBox -calcBut: Button -timeLab: JLabel -ingGroup: ButtonGroup -RecipesAr: ArrayList
+MainGui() +setCalcs(Recipe):void +resetInput():void -calcButActionPerformed(ActionEvent):void -submitButActionPerformed(ActionEvent):void -exportMenuActionPerformed(ActionEvent):void -exportRecActionPerformed(ActionEvent):void -exportRecActionPerformed(ActionEvent):void -addingActionPerformed(ActionEvent):void -removeIngPerformed(ActionEvent):void -removeRecPerformed(ActionEvent):void

Recipes
-calories:int -name:String -price:double -servings:double -components:String -notes:String
+Recipes() +Recipes(calories:int, name:String, price:double, servings:double, components:String, notes:String) +setName(name:String):void +setPrice(price:double):void +setServings(servings:double):void +setNotes(notes:String):void +setComponents(components:String):void +setCalories(calories:int):void +getCalories():double +getPrice():double +getNotes():String +getName():String +getServings():double +getComponents():String

SearchSort
-index:int -min:int -max:int -key:String
+SearchSort() +sortTablePrice: Recipes[0...15] +searchName(key:String):Recipes +sortCalories():Recipes[0...15]

Calculations
-calcProfits(Recipes):void -calcTime(Recipes):void - -key:String
+SearchSort() +sortTablePrice: Recipes[0...15] +searchName(key:String):Recipes +sortCalories():Recipes[0...15]