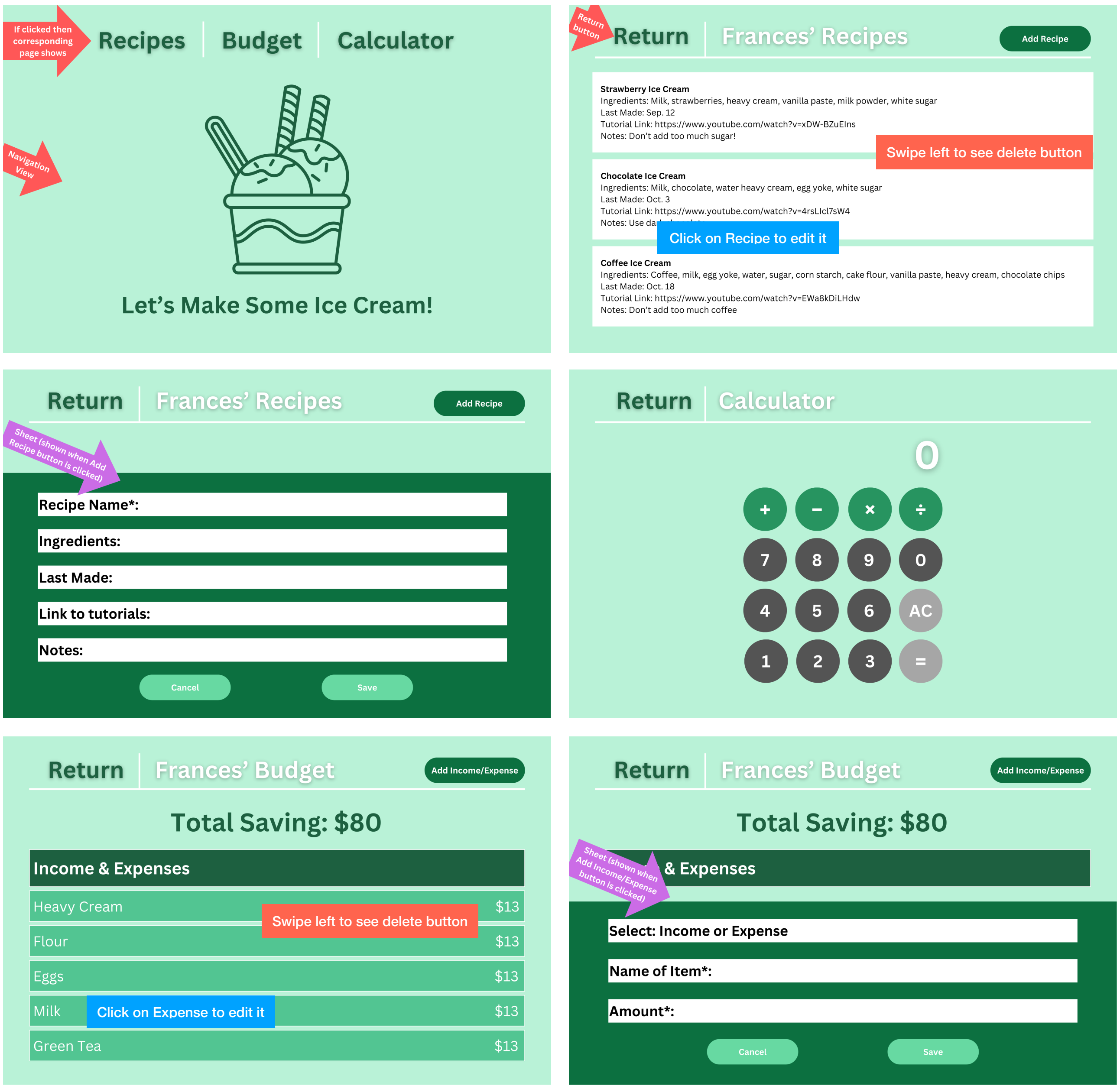
Criterion B: Solution Overview

Design Overview

There will be a home page, linked by buttons to three other pages: Recipes, Calculator, and Budget. The user can view, add, edit, and delete recipes on the Recipes page, compute integer calculations with the four basic arithmetic operations on the Calculator page, and view, add, and delete expenses and income on the Budget page. The return button takes user back to home page. (Items with * means the field is mandatory to fill.) (Figure 1)



Strawberry Ice Cream
Ingredients: Milk, strawberries, heavy cream, vanilla paste, milk powder, white sugar
Last Made: Sep. 12
Tutorial Link: https://www.youtube.com/watch?v=xDW-BZuEIns
Notes: Don't add too much sugar!

Chocolate Ice Cream
Ingredients: Milk, chocolate, water heavy cream, egg yoke, white sugar
Last Made: Oct. 3
Tutorial Link: https://www.youtube.com/watch?v=4rsLIcl7sW4
Notes: Use da

Coffee Ice Cream
Ingredients: Coffee, milk, egg yoke, water, sugar, corn starch, cake flour, vanilla paste, heavy cream, chocolate chips
Last Made: Oct. 18
Tutorial Link: https://www.youtube.com/watch?v=EWa8kDiLHdw
Notes: Don't add too much coffee

Figure 1: User Interface Design (created by kdm500)

Navigation buttons on home page:

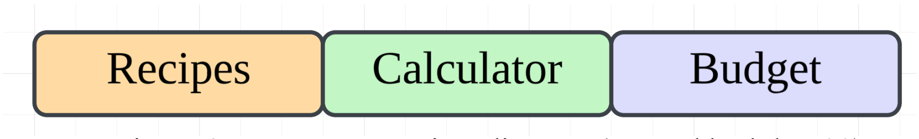


Figure 2: Program overview diagram (created by kdm500)

Sheets (a “pop up” view in SwiftUI) and functions of each page (shown after the page’s button on the home page is clicked):

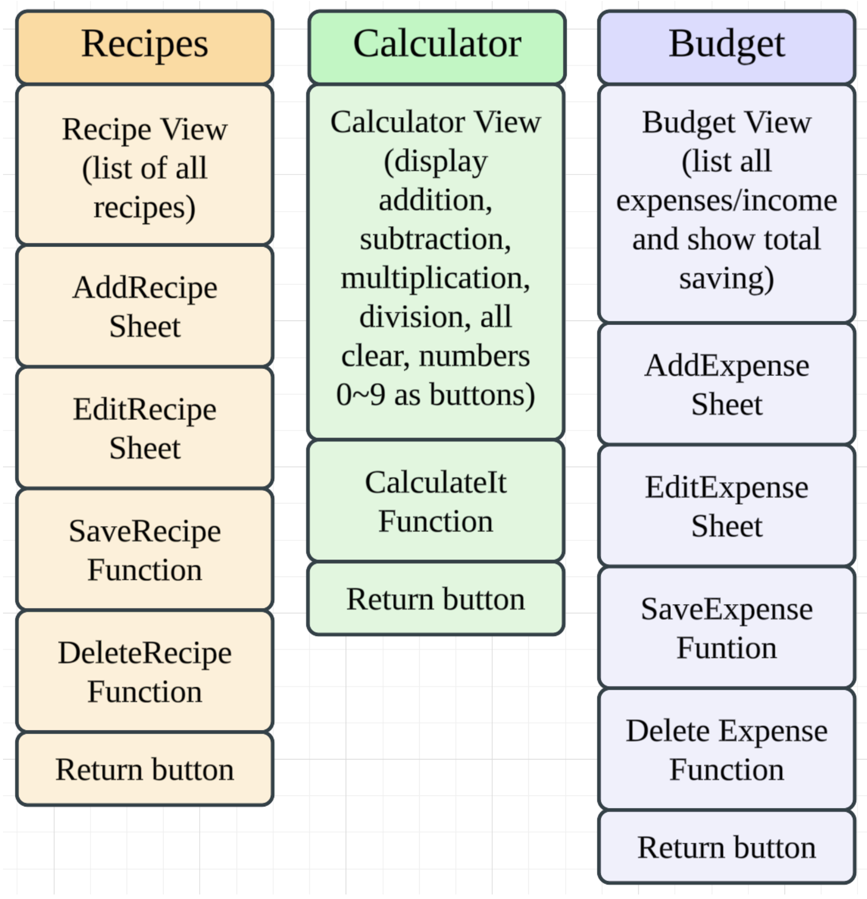


Figure 3: Program overview diagram (created by kdm500)

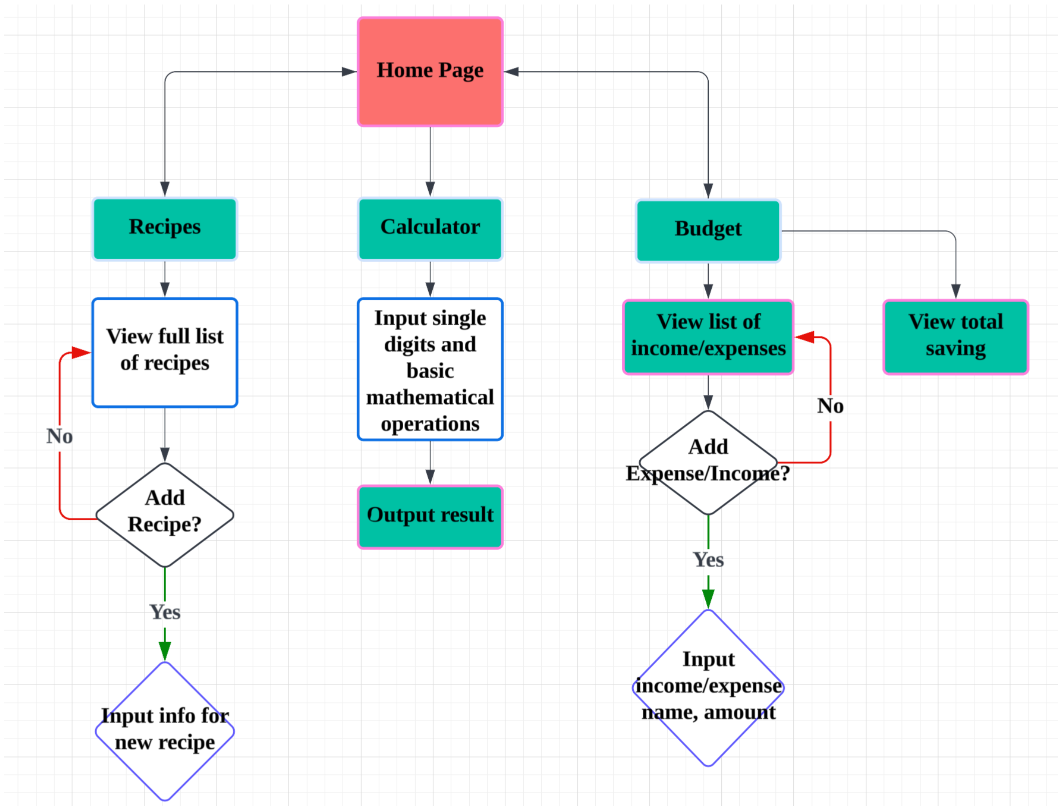
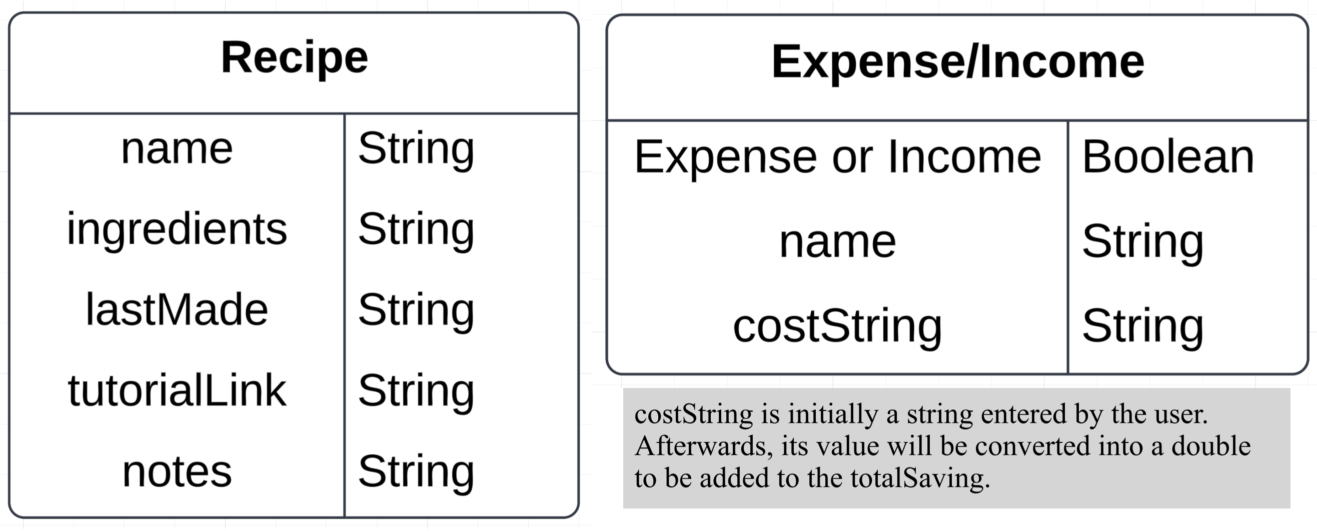

Home Page
Recipes
Calculator
Budget
View full list of recipes
No
Add Recipe?
Yes
Input info for new recipe
Input single digits and basic mathematical operations
Output result
View list of income/expenses
No
Add Expense/Income?
Yes
Input income/expense name, amount
View total saving

Figure 4: Pseudo-algorithm diagram (created by kdm500)

Variables and data types of Recipe and Budget views:



Recipe	
name	String
ingredients	String
lastMade	String
tutorialLink	String
notes	String

Expense/Income	
Expense or Income	Boolean
name	String
costString	String

costString is initially a string entered by the user. Afterwards, its value will be converted into a double to be added to the totalSaving.

Figure 5: UML diagram for variables (created by kdm500)

Pseudocode: (When adding a new recipe; also applies to adding a new expense/income) (Also see Figure 6)

```
if !("NAME".isEmpty) AND ("Save" button is clicked) then
        Call saveRecipe() function
end if


if "Add" button is clicked then
        Show AddExpense sheet
        if !Name.isEmpty AND !Amount.isEmpty AND
"Save" button is clicked then
                Call SaveRecipe function
        end If
end if
```

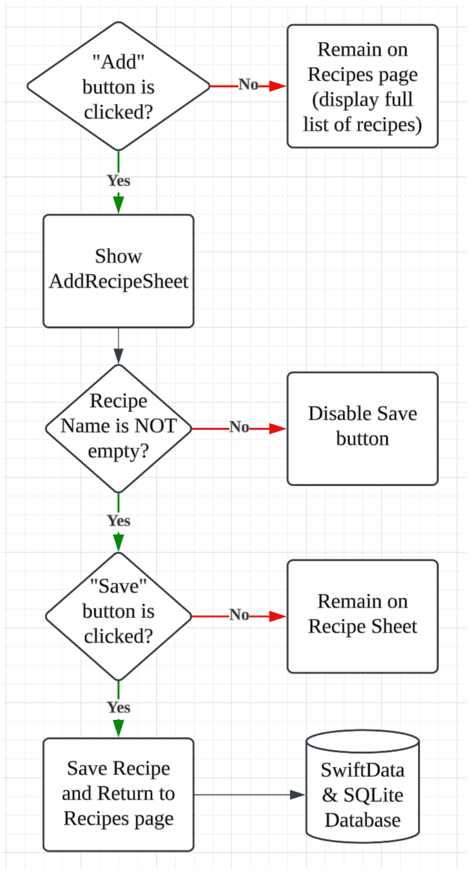

"Add" button is clicked?
No
Remain on Recipes page (display full list of recipes)
Yes
Show AddRecipeSheet
Recipe Name is NOT empty?
No
Disable Save button
Yes
"Save" button is clicked?
No
Remain on Recipe Sheet
Yes
Save Recipe and Return to Recipes page
SwiftData & SQLite Database

Figure 6: Pseudo-algorithm diagram for adding a Recipe (created by kdm500)

Test Plan

Test Actions	Success Criteria	Test Methodology	Expected Result
Client can click on buttons to navigate between the pages	Functionality (1)	Go to the Recipe/ Calculator/Budget page by clicking on the their respective buttons, and then press the return button to return to Home page	When a button of a page on the Home page is pressed, the corresponding page shows. When the return button is pressed, the Home page shows again.
Client can add a new recipe on the Recipes page (they can input the name of the recipe, the ingredients, date of the last time it was made, links to tutorials, as well as notes, per the client's request)	Functionality (2, 3)	Try adding a new recipe on the Recipes page	A new recipe is successfully added to the Recipes page. Entries with no names will not be saved. New recipe is saved in app (even when closed and quitted). Details of each recipe remains unchanged unless edited by the user.
Client can edit and delete an existing recipe	Functionality (2, 3)	Try clicking on the name of the recipe to display the EditRecipe sheet and then edit and save the recipe. Try swiping on a recipe and clicking the delete button to delete it.	The recipe can be edited through clicking on the recipe's name. The recipe is deleted when the red delete button is clicked.
On the calculator page, client can input numbers and operations and calculator will output accurate results	Functionality (4, 5)	Go to calculator page, input numbers and operators in calculator, and look at outputted results	The calculator successfully and accurately outputs the results of simple arithmetic operations
Test for extreme values (calculator and budget)	Functionality (4, 5, 6, 7)	Calculator: Try multiplying values of 7+ digits (although the client mentioned that most of her calculations are of values with under 5 digits) Budget: Try entering strings (with chars/numbers/signs/ symbols), large values, and values with more than 2 decimal digits (not just doubles).	Calculator: The maximum value acceptable by SwiftUI and SwiftData has 19-digits. Anything within 19-digits should work as expected. Anything beyond that might lead to an overflow. Budget: If a string is entered, the amount displayed should be automatically defaulted to 0.00. If more than 2 decimal places are entered, only 2 should be excepted (with no rounding).
Client can input income and expenses on Budget page	Functionality (6, 7)	Go to Budget page, click on "Add" button. Input name of item and amount saved or spent.	A new income/expense is successfully added to the Budget page. No empty entries will be saved. New income/expense is saved in app (even when closed and quit). Detail of each income/ expense remains unchanged unless edited by the user.
Client can delete existing income and expenses on Budget page	Functionality (6, 7)	Try swiping on an expense and clicking the delete button to delete it.	The income/expense is deleted when the red delete button is clicked.
Client can see total saving (plus/minus all past income/ expenses) on Budget page	Functionality (7, 8)	Look for total saving on Budget page	Total saving is readily available on Budget page

Test Actions	Success Criteria	Test Methodology	Expected Result
App functions without WiFi	Functionality	Turn off WiFi on device/ MacBook and open app. Try adding Recipes or Expenses to the app.	The app itself runs, its functions work, and data is saved without WiFi connection.
App uses a color scheme of black, white, and green (different shades but same hue/ saturation).	Design (8, 9)	Observe the colors used for all pages of the app.	App uses predominantly black, white, and green colors.
Buttons are clearly highlighted using colors, shade, and/or shape)	Design (8, 9)	Observe the buttons' colors, shades, or shapes.	Buttons are clearly indicated by their color, shade, and/or shape.