DishDoc

A recipe management solution

By: Amrita, Aryan, Evan, Aima, Shivank, Riley, and Theodore

Table of contents

01 Objective

02 Requirements

03 Function Point Analysis

04 Project Timeline

05 Cost Estimation

06 Use Case Diagram

Table of contents

07 Sequence Diagrams

08 Class Diagram

09 Architectural Design

10 Competitor Comparison

Objectives

- We will be implementing a recipe manager website.
- Users will be able to create an account or login in order to create, view, edit, or delete their recipes.
- Every recipe will have a list of ingredients along with the instructions.
- Additionally, recipes will be able to be searched for by name, ingredients, or various tags that can be given to each recipe (breakfast, lunch, dinner, quick, instant pot, etc).
- Users will also be able to take a list of recipes and generate a shopping list from their ingredients, summing up ingredients that are used in multiple recipes.

Requirements

Functional

- The system shall enable users to login securely using unique credentials.
- The system shall allow authenticated users to create recipes.
- The system shall enable users to manage tags for categorizing their recipes.
- The system shall allow users to search for recipes by different criteria (e.g., “breakfast,”“lunch,”“dinner,”“chicken”).
- The system shall automatically sum ingredients from multiple recipes in the shopping list.

Non-Functional

- The system shall provide an intuitive interface that allows users to complete any function within five clicks from the homepage.
- The system shall return recipe search results within 2 seconds.
- The system shall be available 95% of the time.
- The system shall operate across all major browsers (Chrome, Firefox, Edge, Safari).

Function Point Analysis

	Category	Count	Complexity			Count * Complexity
			Simple	Average	Complex	
1	Number of user inputs	12	**3**	4	6	36
2	Number of user outputs	4	4	**5**	7	20
3	Number of user queries	3	**3**	4	6	9
4	Number of data files	5	7	**10**	15	50
5	Number of external interfaces	1	**5**	7	10	5
					GFP	120

Function Point Analysis

PC	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Score	4	4	2	2	1	4	3	4	3	2	2	3	1	5

Assumptions:
Team size = 3 developers
Efficiency = 15 fp/person-week

Calculations:
PCA = 0.65 + 0.01(4+4+2+2+1+4+3+4+3+2+2+3+1+5) = 1.05
FP = GFP * PCA = 120 * 1.05 = 126
E = PCA / Efficiency = 126 / 5 = 8.4 person-weeks
D = E / Team size = 8.4 / 3 = **2.8 weeks**

Project Timeline

With 3 developers, we expect this project to take approximately three weeks, from December 1st to December 19th, with developers working eight hours per day, excluding weekends. Throughout this period, we will employ an iterative design model to develop our software.

Iterable 1: Deliver a Functional Recipe Manager App (1.5-2 Weeks)

- Front end with pages for user information, all recipes, one recipe, and a shopping list
- Database for storing user information, recipes, ingredients, and shopping lists
- Ability to perform CRUD actions on user information, recipes, and shopping lists from the front end

Project Timeline

Iterable 2: Linking Recipes to Shopping List (0.5-1 Week)

- Able to automatically add ingredients from a recipe to the shopping list
- Shopping list gets simplified so that repeated ingredients are merged with their amounts converted into the same units and combined

Iterable 3: Better Organization (0.5 Weeks)

- Recipes can now be tagged using a variety of preset or custom tags
- Recipes can be searched with multiple queries such as name, tag, ingredients, cooking time
- Previously viewed or commonly viewed recipes are remembered and displayed earlier in the list
- Estimated Total: 2.5-3.5 Weeks

Cost Estimation

Hardware Products

Laptops: $1,000/dev = $3,000
Total: $3,000

Personnel

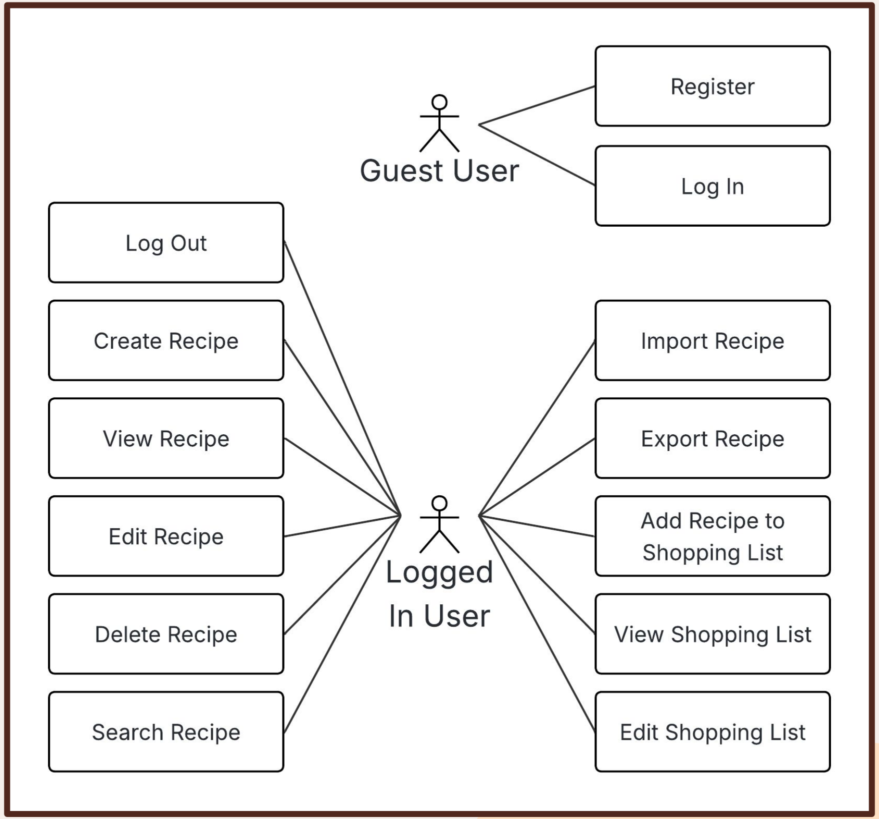
Developer: $40/hr = $12,600
Total: ~$12,600

Software Products

IDE: Free
Claude Pro Subscription: $20/dev = $60
Tech Stack (React, Node.js, Express): Free
Database: $60/month = $720/year
Cloud Hosting: $20/month = $240/year
Domain: $20/year
Total: $60 + $980/year

Total: $15,660 + $980/year

Use Case Diagram



Guest User

Register

Log In

Logged In User

Log Out

Create Recipe

View Recipe

Edit Recipe

Delete Recipe

Search Recipe

Import Recipe

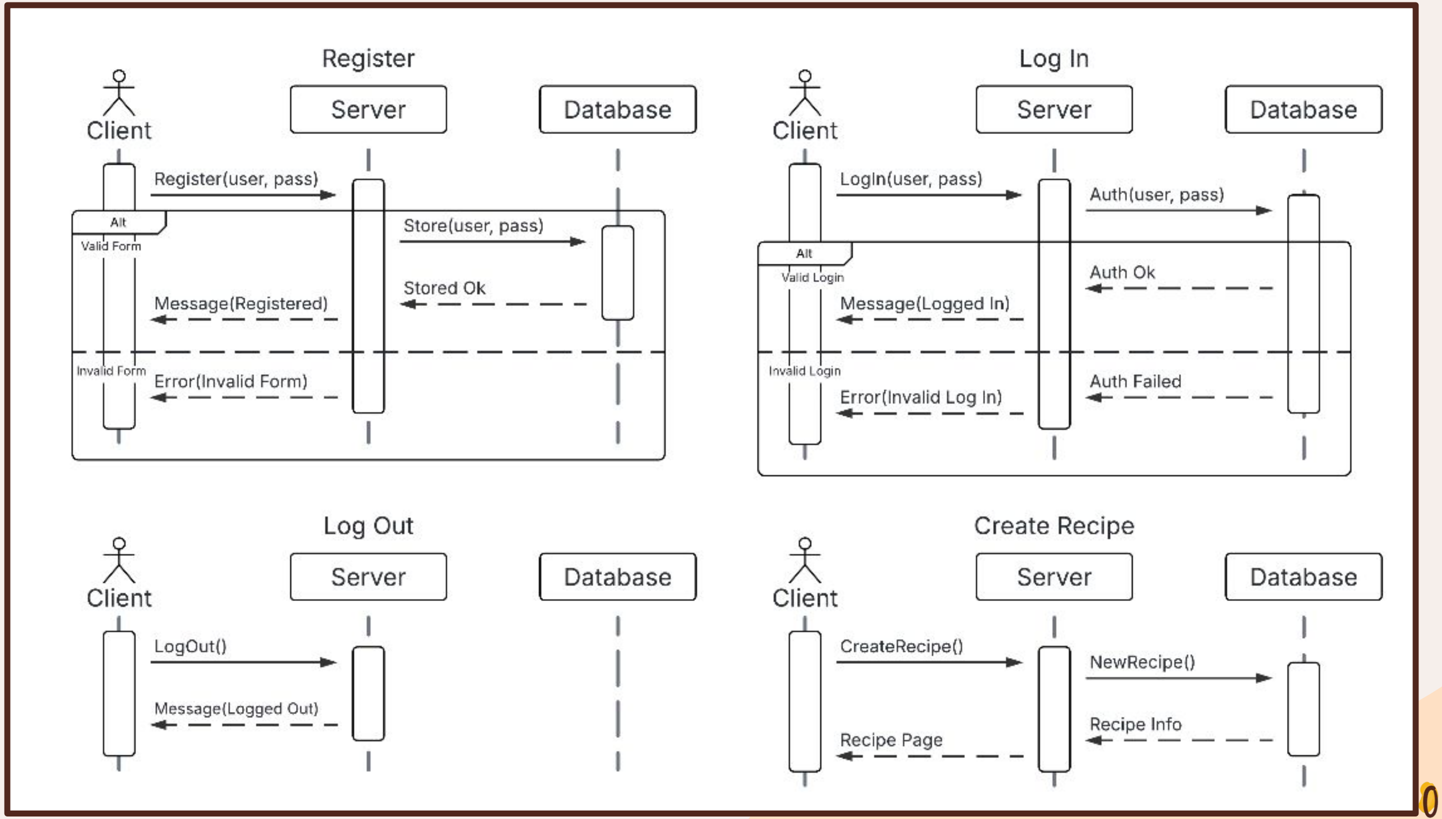
Export Recipe

Add Recipe to Shopping List

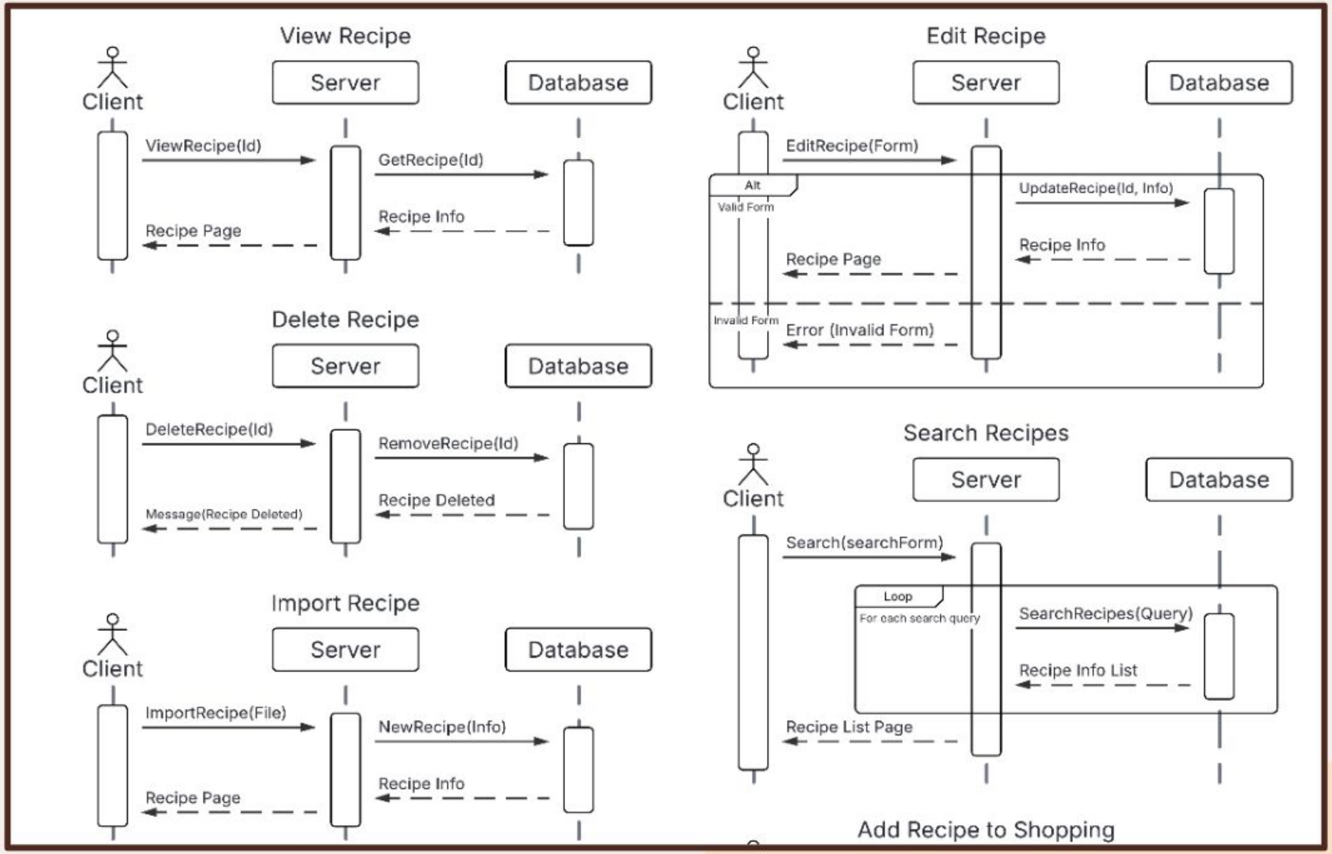
View Shopping List

Edit Shopping List

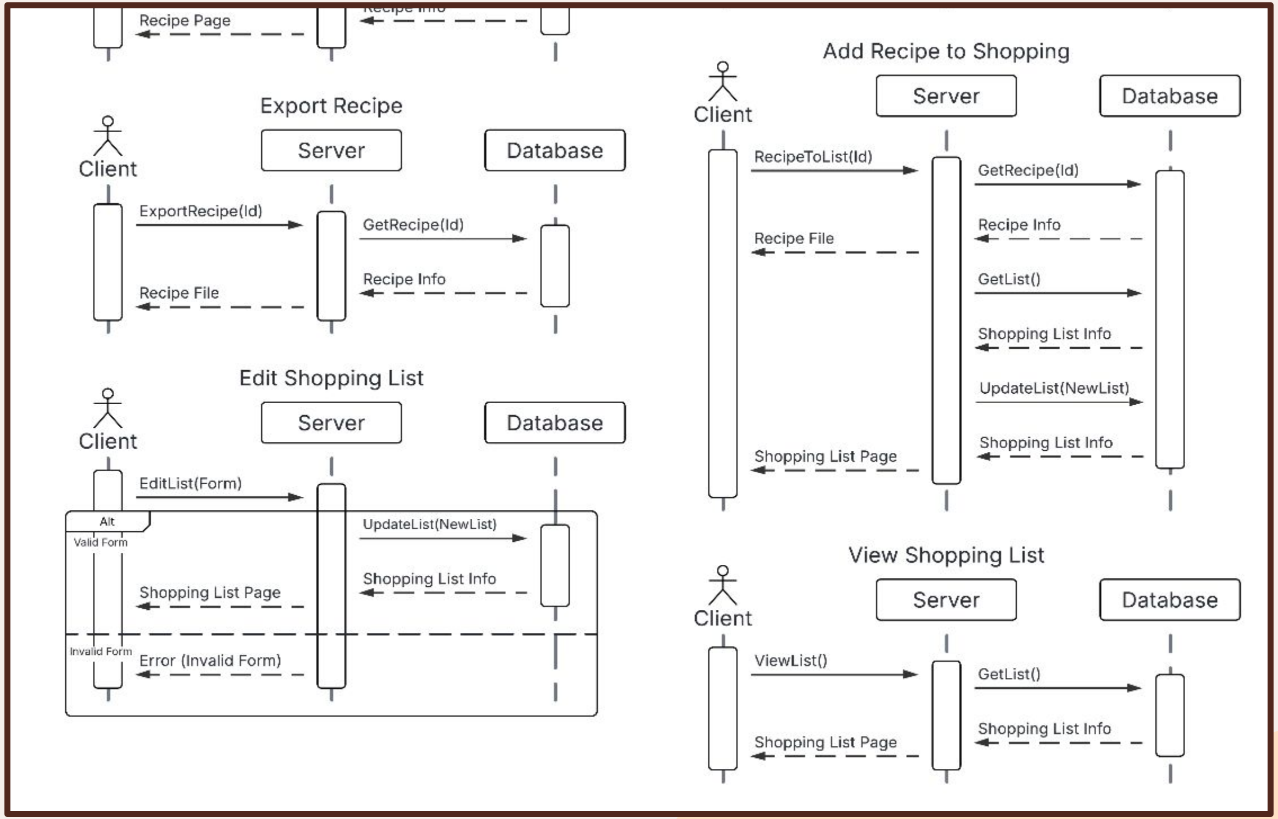
Sequence Diagram (1)



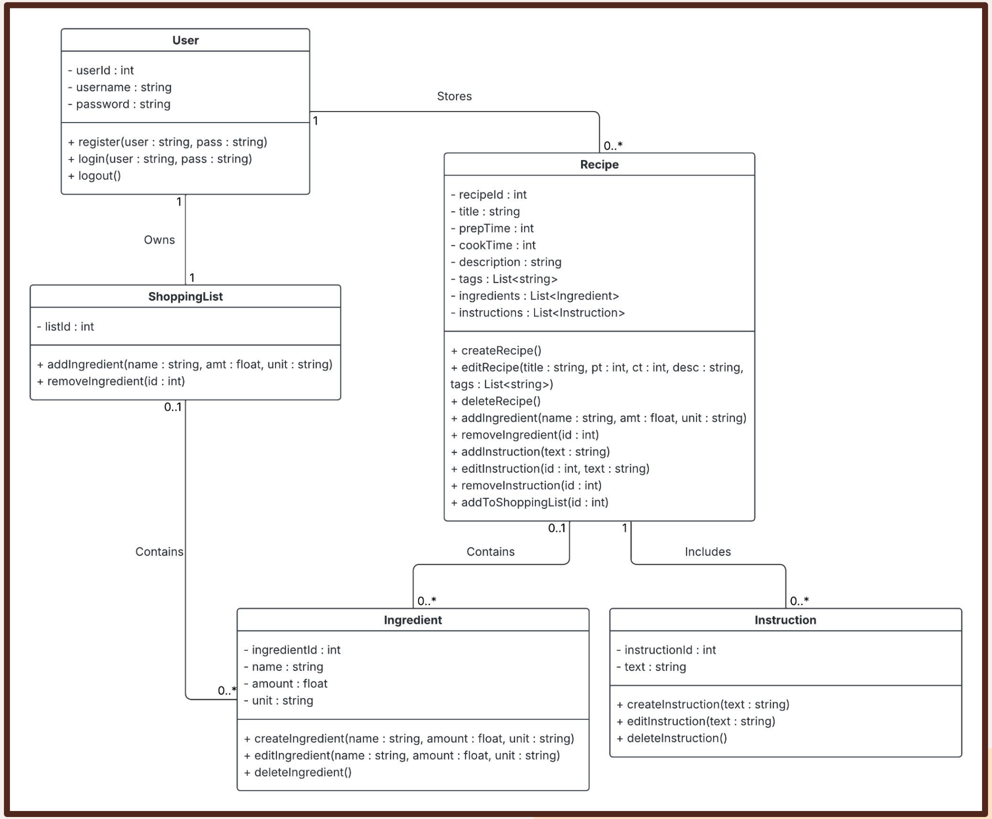
Sequence Diagram (2)



Sequence Diagram (3)



Class Diagram



User

- userId : int
- username : string
- password : string

+ register(user : string, pass : string)
+ login(user : string, pass : string)
+ logout()

Recipe

- recipeId : int
- title : string
- prepTime : int
- cookTime : int
- description : string
- tags : List<string>
- ingredients : List<Ingredient>
- instructions : List<Instruction>

+ createRecipe()
+ editRecipe(title : string, pt : int, ct : int, desc : string, tags : List<string>)
+ deleteRecipe()
+ addIngredient(name : string, amt : float, unit : string)
+ removeIngredient(id : int)
+ addInstruction(text : string)
+ editInstruction(id : int, text : string)
+ removeInstruction(id : int)
+ addToShoppingList(id : int)

ShoppingList

- listId : int

+ addIngredient(name : string, amt : float, unit : string)
+ removeIngredient(id : int)

Ingredient

- ingredientId : int
- name : string
- amount : float
- unit : string

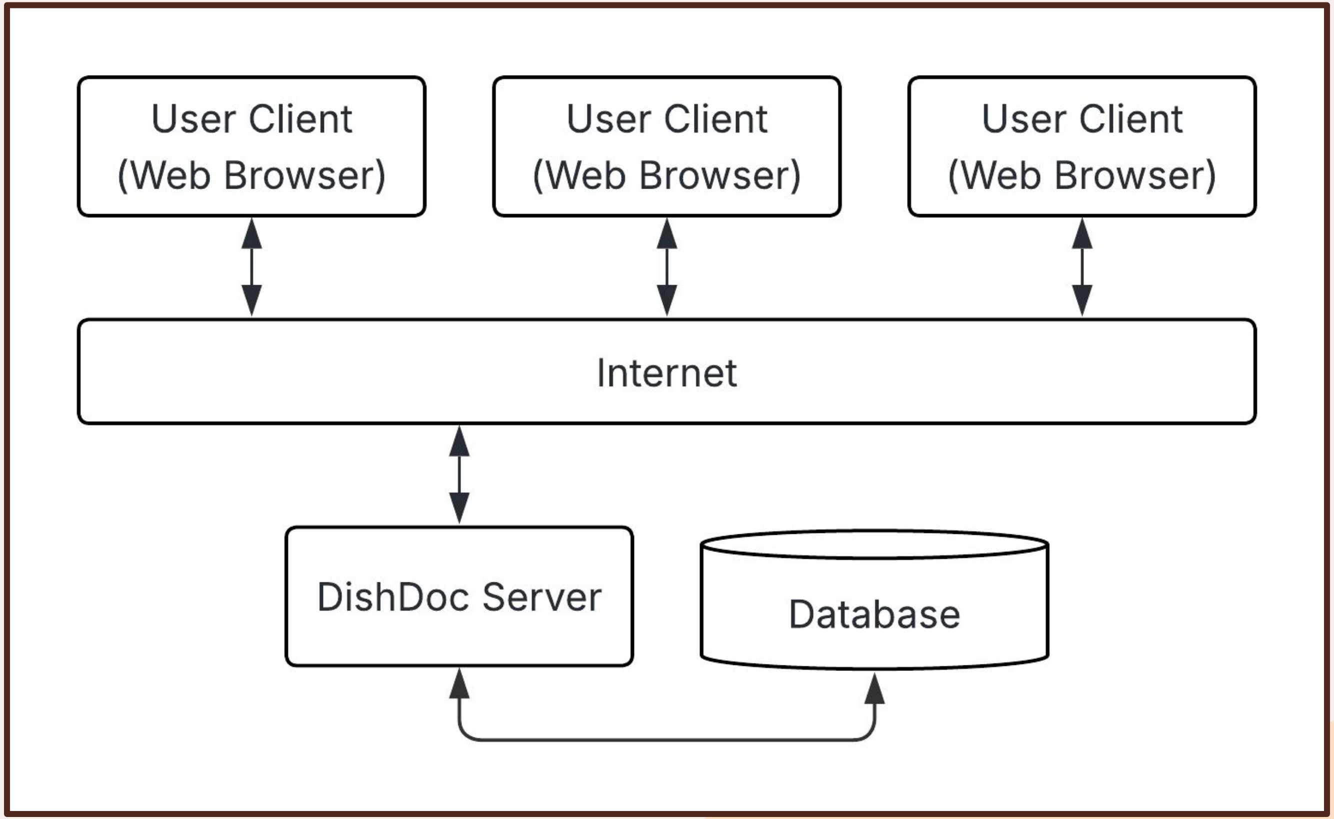
+ createIngredient(name : string, amount : float, unit : string)
+ editIngredient(name : string, amount : float, unit : string)
+ deleteIngredient()

Instruction

- instructionId : int
- text : string

+ createInstruction(text : string)
+ editInstruction(text : string)
+ deleteInstruction()

Architectural Design



Competitor Comparison

DishDoc focuses on minimizing user friction and maximizing utility, solving pain points that established competitors like Tandoor, Paprika, and Recipe Keeper overlook.

- Shopping List Logic
  - Competitors: Basic aggregation (listing items multiple times)
  - DishDoc: Advanced summation and unit conversion
- Organization and Search
  - Competitors: Limited to fixed categories
  - DishDoc: Prioritizes user control over categories with custom tags
- User Interface
  - Competitors: Dated interfaces that feel cluttered / heavy
  - DishDoc: Built on modern web stack to be cleaner, aesthetic, and highly intuitive

References

- [1]Jest, “Jest · 🃏 Delightful JavaScript Testing,” Jestjs.io, 2017. https://jestjs.io/

- [2]“Paprika Recipe Manager for iOS, Mac, Android, and Windows,” www.paprikaapp.com. https://www.paprikaapp.com/

- [3]“Tandoor: Smart recipe management,” Tandoor.dev, 2025. https://tandoor.dev (accessed Nov. 21, 2025).

- [4]“Recipe Keeper App for iPhone, iPad, Android, Windows and Mac,” recipekeeperonline.com. https://recipekeeperonline.com/

Thanks!

Any questions?

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**