



# Low Level Design Document

## Introduction

This Low Level Design (LLD) document outlines the implementation details for **RecipeBox - Recipe Organizer**, an open-source web application for users to add, edit, and categorize recipes, including ingredient and step management. The application is built using React, JavaScript, HTML, CSS, and Tailwind.

## 1. System Components

Component	Description	Key Responsibilities
RecipeList	Displays list of recipes	List, filter, and select recipes
RecipeForm	Add/edit recipe details	Input, validate, and submit data
RecipeDetail	Shows full recipe info	Display ingredients and steps
CategoryFilter	Filter recipes by category	Filter logic, UI for categories
IngredientManager	Manage ingredients within a recipe	Add, edit, remove ingredients
StepManager	Manage preparation steps	Add, edit, reorder, remove steps
StorageService	Local storage abstraction	Persist and retrieve recipes

## 2. Class/Interface Overview

Class/Component	Relationships	Key Methods/Props
RecipeList	Uses RecipeDetail	onSelect(recipeId)
RecipeForm	Uses IngredientManager , StepManager	onSubmit(recipeData)
RecipeDetail	Receives recipe prop	-
CategoryFilter	Emits filter event	onFilter(category)
StorageService	Singleton utility	getRecipes() , saveRecipe() , deleteRecipe()

### Example: Recipe Data Model

```
{
  id: string,
  title: string,
  category: string,
  ingredients: [{ name: string, quantity: string }],
}
```



```
    steps: [string]
  }
```

### 3. Data Structure Overview

Model	Fields
Recipe	id, title, category, ingredients[], steps[]
Ingredient	name, quantity
Category	name (string)

### 4. Algorithms/Logic

#### Recipe Add/Edit Flow (Pseudocode):

```
onSubmit(recipeData):
  if validate(recipeData):
    if recipeData.id exists:
      update recipe in storage
    else:
      generate id
      add recipe to storage
      refresh recipe list
  else:
    show validation errors
```

#### Filtering Recipes:

```
filterRecipes(category):
  return allRecipes.filter(r => r.category === category)
```

### 5. Error Handling

Scenario	Handling Approach
Invalid input (empty fields)	Show inline validation errors
Storage failure (localStorage)	Show error message, retry option
Duplicate recipe title	Prevent save, show error
Missing recipe (on select)	Show "Recipe not found" message

End of Document