

Recipe Finder Ionic App - Innovations

Student Number: G00439372

This document describes the innovations and extra functionality implemented in the Recipe Finder application, exceeding the basic requirements outlined in the project specification. A total of 13 innovations have been implemented across various aspects of the application.

Innovation 1: Toast Notifications

File: recipe-details.page.ts

Implemented visual feedback using Ionic's ToastController when users add or remove recipes from their favourites. When a recipe is added, a green success toast appears with a filled heart icon and the message 'Added to Favourites!'. When removed, a yellow warning toast appears with an outline heart icon and the message 'Removed from Favourites'. The toast automatically dismisses after 2 seconds.

Innovation 2: Recipe Count Display

Files: home.page.ts, home.page.html, home.page.scss

After performing a search, the application displays a message showing the number of recipes found, such as 'Found 10 recipe(s)'. This provides immediate feedback to users about their search results before they scroll through the recipe cards.

Innovation 3: Cooking Time Display

Files: recipe.service.ts, home.page.ts, home.page.html

Each recipe card on the home page displays the cooking/preparation time with a clock icon. This was achieved by adding 'addRecipeInformation=true' to the API request, which returns additional recipe metadata including 'readyInMinutes'. The time is displayed as '25 mins' with a blue time-outline icon.

Innovation 4: Loading Spinners

Files: home.page.ts/html, recipe-details.page.ts/html

Visual loading indicators (ion-spinner) are displayed while waiting for API responses. This provides clear feedback that the application is working, improving user experience especially on slower connections.

Innovation 5: Error Handling with User Feedback

Files: home.page.ts, home.page.html

If the API call fails (network error, API limit reached, etc.), a user-friendly error message is displayed in red text: 'Failed to search recipes. Please try again.'

Innovation 6: Empty State Messages

File: favourites.page.html

When the favourites list is empty, instead of showing a blank page, a helpful message is displayed: 'No favourite recipes yet. Add recipes to your favourites from the recipe details page.'

Innovation 7: Dynamic Favourite Button

Files: recipe-details.page.ts, recipe-details.page.html

The favourites button dynamically changes its appearance based on the current state. When a recipe is not a favourite, it shows 'ADD TO FAVOURITES' with an outline heart icon. When it is a favourite, it shows 'REMOVE FROM FAVOURITES' with a filled heart and red color.

Innovation 8: Separated Service Classes

Files: recipe.service.ts, storage.service.ts

The application follows Angular best practices by separating concerns into dedicated services. RecipeService handles all Spoonacular API communication, while StorageService manages all Ionic Storage operations (favourites, settings).

Innovation 9: TypeScript Interfaces

File: recipe.service.ts

Comprehensive TypeScript interfaces are defined for all API data structures: Recipe, RecipeSearchResponse, Ingredient, InstructionStep, AnalyzedInstruction, and RecipeDetails. This provides type safety and enables IDE auto-completion.

Innovation 10: ionViewWillEnter Lifecycle Hook

File: favourites.page.ts

The Favourites page uses Ionic's ionViewWillEnter lifecycle hook. Unlike ngOnInit which runs once, ionViewWillEnter runs every time the page becomes visible, ensuring the favourites list refreshes after removing a favourite from the details page.

Innovation 11: Responsive Image Styling

Files: home.page.scss, recipe-details.page.scss

CSS styling ensures recipe images display correctly across different screen sizes using 'object-fit: cover' to maintain aspect ratio, 'width: 100%' for responsive sizing, and appropriate border-radius for ingredient thumbnails.

Innovation 12: Instructions Fallback Display

File: recipe-details.page.html

The recipe details page handles both structured (analyzedInstructions) and unstructured (instructions) formats from the API. If analyzedInstructions exists, it displays numbered steps. If not available, it falls back to rendering the raw HTML instructions.

Innovation 13: Duplicate Prevention in Favourites

File: storage.service.ts

The addFavourite() method checks if a recipe already exists in the favourites array before adding it. This prevents duplicate entries using the recipe ID for comparison.

Summary of Innovations

#	Innovation	Primary File(s)
1	Toast Notifications	recipe-details.page.ts
2	Recipe Count Display	home.page.ts/html
3	Cooking Time Display	recipe.service.ts, home.page.ts/html
4	Loading Spinners	home.page.ts/html, recipe-details.page.ts/html
5	Error Handling	home.page.ts/html
6	Empty State Messages	favourites.page.html
7	Dynamic Favourite Button	recipe-details.page.ts/html
8	Separated Service Classes	recipe.service.ts, storage.service.ts
9	TypeScript Interfaces	recipe.service.ts
10	ionViewWillEnter Hook	favourites.page.ts
11	Responsive Image Styling	home.page.scss, recipe-details.page.scss
12	Instructions Fallback	recipe-details.page.html

13	Duplicate Prevention	storage.service.ts
----	----------------------	--------------------

These innovations demonstrate understanding of Angular/Ionic best practices, user experience design, TypeScript programming, and mobile application development principles.