Connor de Bruyn, debruconn, 300499203

App Description

What is the app and what is its one main purpose?

“Recipog” is a recipe providing app that allows the user to input ingredients from their pantry and then be provided with food options they can make using the ingredients they have. Its main purpose is to simplify the process of deciding what to cook and provide creative alternatives to cooking for everyday users.

What one problem/concern does it solve?

Lack of creativity when it comes to deciding what to cook. Many people have set recipes that they know how to cook and will not explore out of their comfort zone by cooking new recipes. By having an app that provides user-specific recipes based on what ingredients they currently have, this enables new recipes to be learnt and made. As well as reducing frequent trips to the grocery store to pick up ingredients.

What are the main features of the app?

- An ingredient list that allows the user to select all the ingredients from their pantry.
- A list of recipes tailored towards the ingredients the user has selected.
- A list of liked recipes that contains all recipes the user “liked” presumably after trying it out and wanting to store for cooking again.
- An editable shopping list that the user can add ingredients to.

Who will use the app?

Recipog is tailored towards everyday users that are aged roughly between 18-60 years old. These users should be willing to cook and have access to their own personal kitchen. These users can be within any cooking skill level as the app will provide a rating into difficulty, so that both more experienced chefs and non-chefs can benefit from the app. The users should have previous experience with a lot of (if not all) the design patterns I am using within the app, making the usability experience more seamless and intuitive.

How long would it take for you to develop the application?

Feature / Task	Sub task	Estimation	Reasoning
Preparation	Learning React	~ 5 hours	I have previous experience, but would like more practice
	Learning Firebase	~ 5 hours	I have previous experience, but would like to trial firestore
Database (Shopping list, Recipes list & Ingredients list)	Setup Realtime DB	~ 1 hour	I have done this previously for my ENGR301 project
	Setup Firestore	~ 1 hour	Should have knowledge after learning it for a previous subtask
	Fill recipes firestore	~ 5 hours	Filling in recipes manually / finding an API will take a while
	Fill ingredient images	~ 3 hours	Images shouldn’t be hard to find, but cropping might be taxing
Setup React	Create empty pages	~ 1 hours	Creating placeholder pages shouldn't take long at all
	Create page routes	~ 3 hours	Not knowledgable on page routing so I need to find/follow a tutorial
	Create components	~ 8 hours	Creating reusable components for all pages should be time consuming
Filling Pages with Content	Recipe list page	~ 4 hours	After creation of the components, fitting and aligning shouldn’t take long
	Single recipe page	~ 5 hours	Probably the hardest page as it needs the most components
	Ingredients page	~ 4 hours	The same reasoning as the recipe list page
	Grocery list page	~ 3 hours	Not as much content than the other pages, should be easier
	Liked recipes page	~ 2 hours	Should be able to copy the recipe list page for majority of content

Connor de Bruyn, debruconn, 300499203

App UX Patterns

Please refer to this list for descriptions of UX patterns I reference throughout the document.

Numbers in my wireframes will correlate to the UX patterns described here (I.e. a 16 in my wireframe will show a grid pattern).

1. **Tab Bars** – Appears at the bottom of the screen as a means of switching across different sections of an app **[1,2]**. The tab bar is strictly used for navigation and contains an appropriate number of tabs (4) to not overwhelm the user.
2. **Search Bars** – Pinned in the header of the app so it is always accessible whilst scrolling through page content. Once the user has inputted in an item, a "Cancel" button will appear to clear search if needed **[1,4].**
3. **Colour** – Imparts vitality into the app. A minimal colour pallet (2-3 complementary colours with neutral (black / white) colours) can be used to draw attention to specific features of the app or to make the app more memorable **[1].**
4. **Scroll Views** – Allows the user to pinch, swipe, flick, drag or tap an area of content to scroll or zoom in. This UX pattern allows lists of texts / components to be displayed at a reasonable size for the user without cluttering the screen **[1].**
5. **Text / Character Input** – After the user presses on a typable area an embedded keyboard / keypad appears that allows the user to input text. This keyboard / keypad disappears after input **[2].**
6. **Page Titles** – Used to give the user a sense of clarity to which page they are on. An enlarged piece of text at the top of the screen that displayed the page name they have navigated to **[2].**
7. **Launch Screen** – A lightweight, fast-booting screen. Giving the user fast response time when opening the app so that they know the app has launched. [1,2,4].
8. **Wait Indicator** – An animated widget to give the user the impression that they are loading / processing something **[2,3,4].** Typically done with a spinning circle accompanied by "Loading" text.
9. **Saving** – A button that informs the user that they are saving progress or content for retrieval later **[2,4,5].**
10. **Proximity** – Creating relationships between elements by the spacing between them. Objects that are placed together are presumed to be related (Gestalt law of grouping) **[5].**
11. **Slideshow** – A means to show numerous images without taking up too much space for additional content. The user can tap or swipe to show different images on the slideshow **[2,5]**.
12. **Button** – Shows the user that an action can be performed by tapping the highlighted area **[1,2,5]**.
13. **Clear Entry** – A button or holding action that allows the user to clear an element, text, or selected items from the app if they deem them unneeded **[2]**.
14. **Sort & Filter** – Gives the user access for more specific queries of searches by narrowing results, or prioritizing results by a selected category **[2]**.
15. **Interstitial Screen** – A screen that is displayed whilst there is a break between pages, commonly used to load components of another page without the screen looking like its frozen and non-interactable **[2]**.
16. **Grid** – Displays a set of selectable images that can fit in the available space. Used to be more space efficient by displaying more content in the app **[2,5]**.

Feature 1: Ingredients List

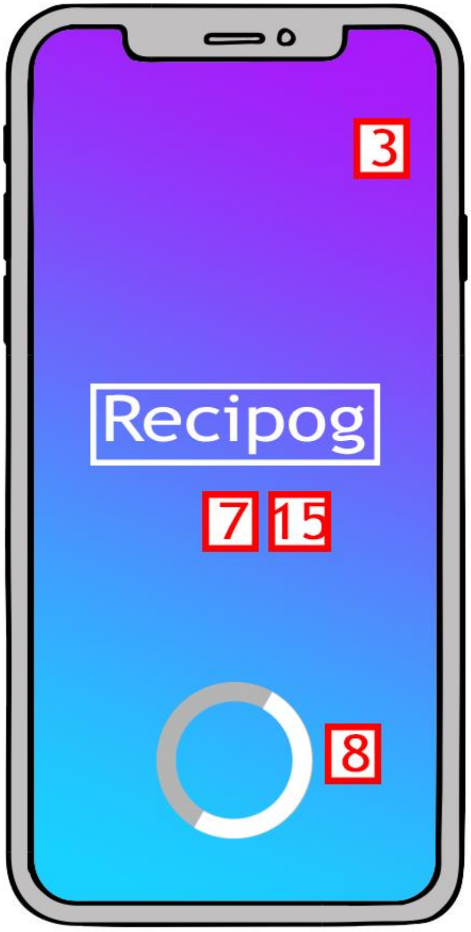


Fig 1: Loading Screen

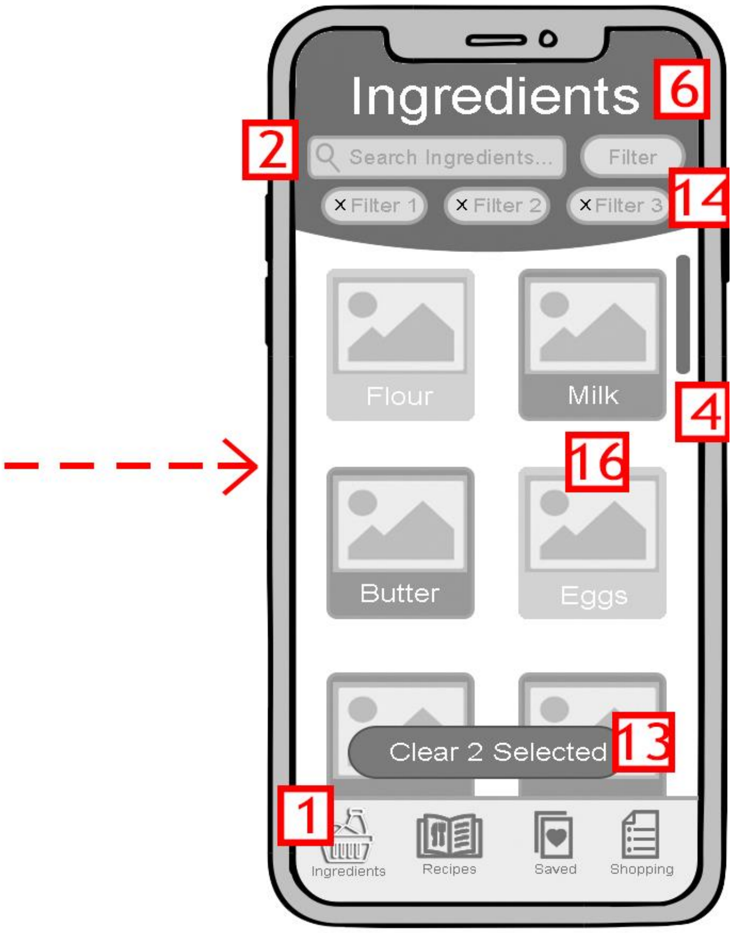


Fig 2: Ingredients Screen

Figure 1: Loading Screen

I have used an **interstitial / launch screen** [1,2,4] for my ingredients list feature as my ingredients screen will be the first screen from launch. The ingredients screen should be the first screen to appear after the interstitial screen as the user should input their ingredients first to follow a logical task flow of this app. Recipes and grocery lists should only be decided after identifying ingredients in the pantry.

A **Colour** [1] gradient has been used to highlight the logo and the **wait indicator,** as well as give a memorable appearance to the app, capturing people's attention to get the users to return and keep using the application.

A **wait indicator** [2,3,4] has been used so that the user doesn't feel like the app is unresponsive after navigating to different pages due to components loading, or slow booting times after launching the app.

Figure 2: Ingredients Screen

In the header for my Ingredients screen I contain a **page title** [2]**.** The users' eyes scan pages from top to bottom, so by including the title at the top they will immediately confirm which page they are on. I also included a **search bar** [1,4] and **sort & filter** [2] design pattern in the header of the screen, allowing an embedded keyboard to show for typing without covering the filters / search bar whilst typing.

A **grid** [2,5] was used to display the ingredients for easy readability. This was used in conjunction with a **scroll view** [1], allowing the user to scroll and tap ingredients to highlight them. The grid provides larger, easier to press buttons with images of appropriate dimensions.

A **clear entry** [2] button is used to clear the selected pantry items if the user wants to restart progress. This button is placed near the bottom of the content pane to avoid mistake taps.

I have included a **tab bar** [1,2] at the bottom of my screen as the primary form of navigation. Using 4 tabs to navigate to the 4 features I listed for my app. By keeping to 4 features, I won't overwhelm the user and make buttons too small for tapping.

Feature 2: Recipes List

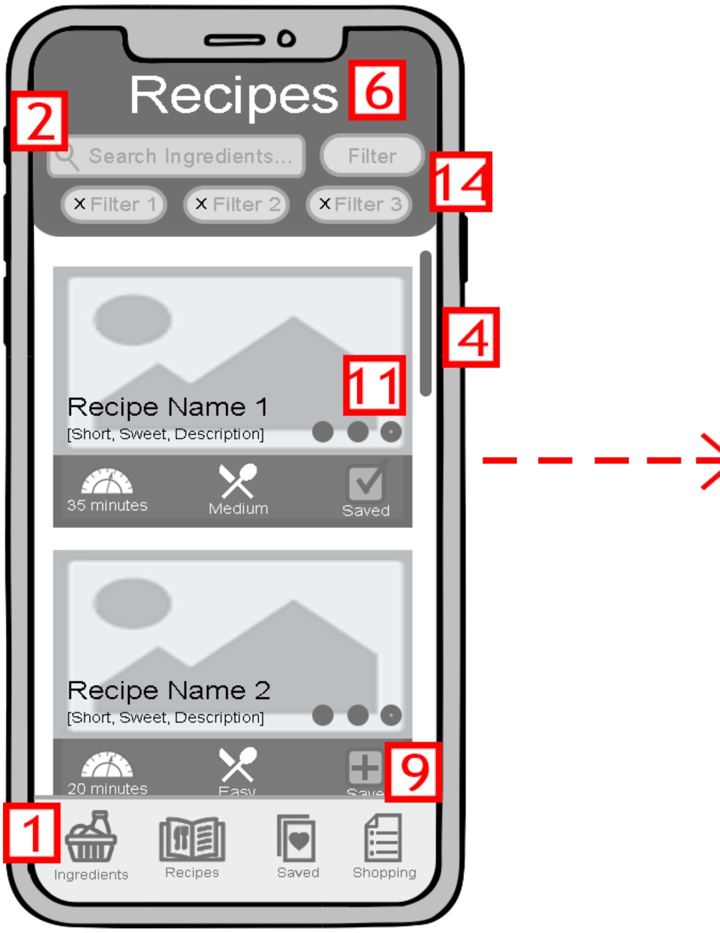


Fig 3: Recipes list Screen

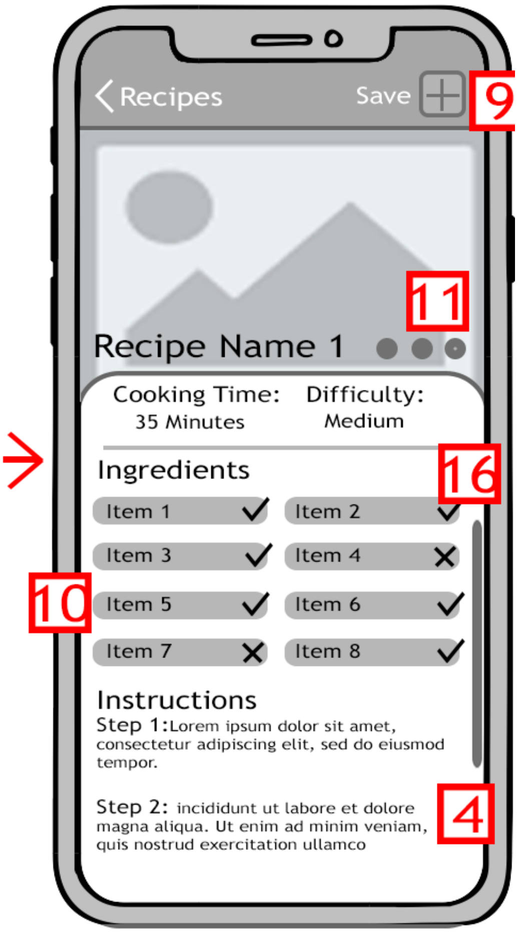

Recipes
Save
9
11
Recipe Name 1
Cooking Time:
35 Minutes
Difficulty:
Medium
Ingredients
16
Item 1
Item 2
Item 3
Item 4
10
Item 5
Item 6
Item 7
Item 8
Instructions
Step 1: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor.
Step 2: incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
4

Fig 4: Recipe Screen

Figure 3: Recipes List Screen

I have used a **page title** [2] at the top of the screen to address to the user which page they are currently on. A **search bar** [1,4] and **sort & filters** [2] button has been used to allow the user to search and filter recipes from the recipes list. These have been placed in the header for to not overlap with the embedded keyboard once inputting text.

Each recipe contains its own **slideshow** [2,5] of images, showing off different recipe images instead of a standard one image, giving the user more context to base their decision of which recipe to cook. The recipes have been placed in a **scroll view** [1]to allow the user to scroll through the recipes list.

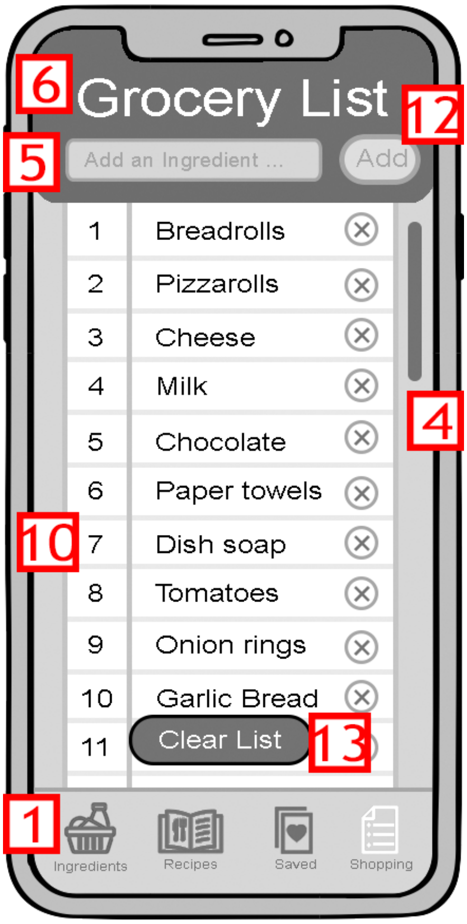
A **saving** [2,4,5] feature has been implemented by pressing the “Add to saved” recipes button. I have included a **tab bar** [1,2] to keep consistent with the other pages and navigation style I have adopted.

Figure 4: Recipe Screen

I have included both the **saving** [2,4,5] and **slideshow** [2,5] design patterns from the previous page. After the user taps a recipe to view more detail and transition to the recipe screen, both the slideshow and save buttons should be more prominent and shown at the top as they are primary functions of the page.

Both the **grid** [2,5] and **proximity** [5] patterns have been used on the ingredients list in the recipe screen to increase space efficiency and to portray relatedness of elements. The content of the recipe has been placed in a **scroll view** [1] to allow for longer recipes and overlap.

Feature 3: Editable Shopping List



A **page title** [2] has been used to keep consistent with the other pages and to confirm they are on the correct page.

A **text / character input** [2] and **button** [1,2,5]design pattern has also been included in the header to allow the user to add ingredients to the shopping list. This has been placed at the top of the screen, so it is easily visible to the user and to not overlap with the embedded keyboard.

Proximity [5] of items in the list have been made intentionally compact to allow for more elements and to give the appearance of an actual grocery list. The items in the list have also been placed in a **scroll view** [1] to allow scrolling in case the list overflows past screen length.

A **clear entry** [2] button is used to clear the grocery list if the user wants to restart entering items from scratch. This has been placed as far down on the content plane to avoid mistake tapping. However still in comfortable reach as it is a primary action the user can take.

The **tab bar** [1,2] is also included to keep navigation consistent amongst pages and to allow navigation back to specific pages the user wants to get to.

References:

1. Apple interface guidelines | https://developer.apple.com/design/human-interface-guidelines/
2. Steven Hoober & Eric Berkman “Designing Mobile Interfaces” | http://4ourth.com/wiki/Designing%20Mobile%20Interfaces
3. Mobile UX patterns library “Pttrns” | https://pttrns.com/

4. Mobile UX patterns library “Mobbin” | https://mobbin.design/browse/ios/apps
5. UI Patterns Design Course “Balsamiq” | https://balsamiq.com/learn/courses/intro-to-ui-design/ui-design-patterns/