Casey Nord, Steven Phillips, Matthew Insley, Adrian Protzel, Brad Goodlett

March 13, 2020

CS 352

Anita Sarma

**Final Prototype**

**Problem Summary:**

We are designing a user interface for an online cooking recipe management system that utilizes decentralized web protocols such as IPFS (<https://ipfs.io/>). From a usability standpoint this focuses on two distinct problems: creating an intuitive and well laid out recipe application, and integrating a web protocol that encourages transparency and direct access to underlying data structures.

As a recipe management system, we are looking to provide an interface that iterates on and improves the design of many websites that exist today. We are striving to eliminate the clutter and poor formatting commonly found on food recipe websites and are hoping to create a functional, user friendly UI that allows users to save, store, share, and access recipes from any device without being tied to a single application. This should provide the everyday user a well formatted and intuitive interface for managing their recipes. Everything should be laid out clearly enough to allow it function as a replacement for recipe books and boxes of notecards, instead of functioning like a blog or a spread out search engine strewn with ads. However, we do know ads are a company's choice to add on a website and we understand this going in.

In terms of web protocols such as IPFS, we are looking to store and manage recipe data in a databaseless, content addressed, decentralized network. Leveraging this technology provides a unique framework where data such as .json files representing a recipe are not stored in a centralized location that is only accessible through a specific website or API. Instead, recipe data is distributed across the entire network, accessible through content identifying hashes. In this way, users will have direct access and control over the data on the system. Since the recipes themselves are stored completely independently of the web app, advanced users are capable of going as far as creating their own recipe tools and applications that are able to recognize, parse, and display the data structures used to represent recipes on the network.

**Users:**

We are targeting two groups of users. The first group is adults and teenage users who are looking online for a better recipe UI online. Specifically one that has good usability, which is why we put everything you need to make a recipe on one page so that users don’t have to scroll with dirty hands. We avoided the pitfalls of ads that clutter most recipe websites. The second user group is IPFS researchers or enthusiasts who want to see a practical application of IPFS technology.

**Justification of design decisions:**

Several updates have been made to the prototype after reviewing the feedback from our heuristic and empirical evaluations, as well as user evaluation from the latest design gallery.

* Links to hash identifiers have been included on all recipe thumbnails. This provides better memory-recognition as accessing these won’t be limited to a single location.
* Question mark buttons have been added in relevant places. This is meant to function alongside a FAQs page that will be implemented to provide documentation and feature specific assistance.
* A standardized footer has been implemented throughout the UI. This adheres to the design principle of consistency and quick access to important information, such as help pages and site information.
* The header has been rearranged and the app title has been made smaller. It now also functions as a link to the homepage. This helps remove unnecessary clutter introduced by using extra buttons when a link on the app title would suffice.
* Profile icons have been moved to the right of the header next to settings. This provides better grouping and helps solidify the functionality of the header.
* ‘Add recipes’ has been replaced ‘create recipes’. This should eliminate confusion by making users have to remember if ‘add’ means to create a recipe or add to the users collection.
* Prototype links have been updated so that things cannot be clicked on when on a confirmation page. This should mirror the actual intended functionality of the app better.
* ‘My favs’ has been removed. The ‘My recipes’ option is enough, and adding additional bins for organizing recipes only makes it more difficult for a user to manage.
* ‘Collapse’ button has been replaced with ‘Return’. This provides better clarification of its purpose and improves navigation and the design principle of consistency across the app.
* The double slider has been replaced with a static bar. This was never intended to be a feature but in the wireframe seems to suggest undesired functionality.
* ‘Add to Favorites’ is now ‘Add to My Recipes’. This provides better consistency with the updated ‘My Recipes’ feature.
* View recipe now links to a ‘Modify Recipe’ page instead of linking to ‘Create Recipe’. This should function the same, but will pre-populate the ‘Add Recipe’ page with data pulled from the recipe the user linked from.
* In the final design, all hash links will be togglable from user settings and will be disabled by default.
* Food categories have been grouped by relevance (eg. protein next to protein).

In total, these changes should better adhere to the design principle, notably visibility, affordance, and consistency.

**Design Principles:**

* Visibility:
  + All the information a user would need is displayed without needing to go to multiple pages.
  + Recipes are grouped into sortable categories.
  + Each recipe has a title with and a thumbnail photo.
  + Categories can be expanded and collapsed without navigating away from the main page.
* Affordability:
  + Scrollbars and buttons are utilized strategically to make app functionality clear and intuitive.
  + Each recipe thumbnail is clickable to take the user to the recipe view page.
  + User recipes and searchable recipes have a different styling to make them distinct.
  + The app title in the header functions as a link to the home page.
  + Hash ID visibility can be toggled allowing advanced users to see more information.
* Consistency:
  + Recipes are displayed similarly in a box, with a title.
  + User recipes and home page buttons are always in the same place.
  + Navbar and footer are consistent across all pages only changing where necessary.
  + Each recipe has a minimum requirement of a title, image, process, and ingredients.
* Feedback:
  + Expanding/collapsing pages can be animated to provide a better visual experience.
  + Prompts are displayed for events such as adding and cancelling recipes.
  + IPFS hashes are displayed if enabled for advanced users.
  + Mouseover tooltips and hover based interaction will give better interactivity.
* Constraints:
  + Each recipe has a minimum requirements before posting
  + The create recipes page uses forms to help enforce proper formatting and data structures.
  + Hash display is initially turned off in profile settings.

**Feedback from prior feedback:**

Most feedback said that our design was very helpful and high functioning. Though a little cluttered, it has all the functions necessary at any given page. As requested, we organized the food categories listed, being more grouped together such as proteins being listed next to each other. The hash identifier has been confusing for some time so we made it toggleable with it being turned off by default. Our site doesn't have too many pages though some redundancy was found with the my favorites and my recipes pages, to fix this we lumped them together.

**Heuristic Evaluation Analysis:**

* Capable error prevention walls will allow us to decrease user errors when adding data on the add recipe screen. We’ve tried to address this in the design from the start by making the **add recipe** screen use individual boxes for each specific piece of data on a recipe. We’ve learned from our evaluation that this needs to be paired with a robust system for handling errors to be effective.
* Need to allow for more user control on the **add recipe** page, possibly when designing their layout of adding a recipe that fits their needs.
* Custom error pages need to be implemented to provide direction in case of errors. These could be overrides of standard errors like 404, as well as purpose built for internal issues we can identify and provide error handling and redirection for. This is mostly to handle potential issues of files not being found when navigating to a **view recipe** page.
* Links to hash identifiers should be included directly on home page thumbnails for ease of access. This shouldn’t add additional clutter to the screens where these will be implemented. Without doing this, identifiers are limited only to the **view recipe** screen, which can prevent power users from grabbing the identifier to query the raw data directly when troubleshooting.
* Question mark help buttons need to be included throughout to summarize specific information regarding app functionality. This would allow help for specific features where those features exist, preventing a user from having to reference a single monolithic faq for all questions. This is applicable on **all pages**.
* A standardized footer needs to be implemented that provides access to, among other things, a faq page that explains website features, as well how it utilizes IPFS as a means to access recipe data without the use of a database (this should also include links to places like <https://ipfs.io/>. This applies to **all pages**.
* Alternative methods of adding data should be considered, such as a more generalized system that can parse a block of text to give an alternate way to add recipes. This applies to the **add recipes** page.
* A guide should be provided to inform interested users about the data structures used and how data can be added to IPFS directly without having to access the web app. This would be a part of the implemented **faq page** that should be implemented in the future.
* Customization options or alternative displays for viewing recipes could be added to augment the user experience. This would apply to the **add recipe** page.

**Insights:**

* Favorites and quick favorites provide good functionality and faster searching for recipes. This should be kept in the layout as we move forward, and seems to be an important feature.
* Different thumbnail designs for favorites and standard recipes makes it easier to recognize which is which. It help s delineate the two, and provides an instant visual cue for what is displayed when the page is expanded for public recipes/quick favorites.
* We were slightly worried that the home page design was too busy, and that the expanding feature may bury information too much. This does not seem to be the case. One hard aspect of this design is that there is a lot of information to display, and there is a balance that needs to be maintained. We seem to be in a fairly good spot right now.
* Documentation, faqs, help buttons will be valuable tools to help explain certain features and technologies used to operate the application. This can be implemented and tested in upcoming mockups.
* We may consider shuffling around the order that add recipe items are displayed. This might be user preference, but we can experiment with this on upcoming user evaluations with alterations of the add recipe webform.
* Adding a scale recipe feature to the view recipe page would be smart.