Design Feedback Response

Bryan Goggin

In this document I will be addressing feedback given from the instructor as well as reflecting on my own feedback about the project.

UI Feedback – Personas

Feedback: Users like "Potluck Patricia" may feel intimidated using an app instead of a paper cookbook.

Response: Indeed, the key will be to make app usage intuitive and simple so that even non-tech-savvy users can enjoy the experience. No changes accepted.

Feedback: "Single Mother Suzanne" is budget conscious, yet nothing about the app related to finance. Why would she be interested in this tool?

Response: Noted. A budget-friendly or "price-checker" option which integrates with nearby grocery stores will make a stong stretch feature. No changes accepted.

UI Feedback – Wireframe

Feedback: The design is ugly and does not look modern.

Response: I agree. However, keeping in mind the MVP, so long as functionality is there for a user to find and submit recipes, UI designs can be enhanced in later versions. Will explore more modern templates if time allows.

Feedback: The "Upload Page" does not exist.

Response: Will add wireframe for upload page.

Feedback: Promoted Content may expand the scope of MVP.

Response: Indeed this is true. Limit scope of promoted content to static images. Consider promoted content and Recipe/Ingredient of the Day as a stretch feature.

UI Feedback – Storyboard

Feedback: Full story board is ambitious for MVP

Response: The core functionality is recipe search and upload. Some pages are not needed to achieve that. For MVP, deprioritize Contact, About, Career pages, as well as My Account and My Cookbook and move these into stretch features or static mock ups.

UI Feedback - Style Guide

Feedback: The Style Guide does not match the mock up

Response: Update Style Guide to match mock up

DB Feedback

Feedback: How will the search by be supported?

Response: "Search by" is a nice-to-have. Can implement later. Move "search by" to stretch feature.

Feedback: Is a no-SQL technology ideal for this project?

Response: Yes. I am firm on this choice after researching no-SQL databases. The non-uniformity of different ingredient lists along with the text-based instructions storage are best served with no-SQL. No changes accepted

Feedback: Is this approach scalable?

Response: Its hard to know the future, but for the MVP, this approach will work. As the app evolves, changes to the back end back can be proposed in an agile manner. No changes accepted, but scalability concern is noted.

Feedback: Given the base DB design, will it be possible to add new tables to support new features in the future?

Response: I believe so. The 2-key index system will support "linking" of elements in different tables, should the need arise. These will function in lieu of proper SQL joins. The fundamental object stored in the tables will be the recipe, and all attributes will relate to 1 or more recipes. No changes accepted.

Service Layer Feedback

Feedback: No error responses are shown.

Response: Include responses for page not found, unauthorized and no results shown, as well as what will be displayed when something is broken on the back end.

Feedback: By taking the “search by” feature out of the MVP, how will the backend find results?

Response: Mock up how the search feature will work when recipe entries have data in multiple structures.

Feedback: There is no endpoint to submit a review.

Response: Opps. Add endpoint to submit recipe review.

Conclusion

Some key elements were left off the original design and will be added into the re-design. As well, some elements here extend the scope of the MVP and may jeopardize core functionality. These items have been moved to stretch features and will be documented in github as such. If time allows before final project deadlines, the stretch features will be prioritized and tackled one-by-one in order of importance and value-added.