

You know the feeling... Your grocery list is long and complex, and the store is perpetually busy. Winding past customers laden with baskets, navigating through employees stocking shelves, all while simply collecting the things you need becomes an arduous encounter fraught with opportunity to overlook needed items. Not to mention the fact that you do your shopping at multiple stores, adding to the complexities of location recall.

ESLA endeavors to address these pitfalls by providing the user with a clear data entry format for list creation that allows the user to categorize items by location (ie: dairy, produce, meat). Taking the idea a step further, the user can create store templates to sort their lists by, adapting the order of categories by the flow of the store as well as personal preference. These templates are stored to allow the user to reorganize a list according to shopping location. I believe that this feature is what differentiates my proposal from other applications, and by offering these features with a clear interface it will quickly overtake other simple list-making applications by virtue of its flexibility and versatility.

While targeted towards grocery shopping lists, ESLA is customizable to allow for any categories and templates the user needs. This means that the sorting can be intuitive to individual systems and adapted to various methods, as broad or as refined as the user prefers. The ESLA application is targeted towards users who appreciate the convenience of a lightning fast shopping trip, those who do their shopping at multiple stores, and/or those who have difficulties consistently locating items or explaining the location of items to others. The application can be used on desktop (to allow for easy typing or more intensive sorting) but is primarily intended to work on portable devices such as tablets and mobile phones, particularly in Shopping Mode.

Future features would include the ability to share lists ('Leslie' sends her husband a detailed list that includes photos she's taken of items to assist in proper selections), and the ability to generate lists based on a recipe list database that the user builds over time ('Sinbad' sees that stew meat is on sale and remembers making a great stew last winter. He searches his recipe lists for Stew and finds three entries. He picks the one that appeals most and the app allows for a 'Pantry Check' (the option to remove already obtained items from the sub-list) before incorporating them into his larger shopping list. These items have been pre-categorized by 'Sinbad' so the app is able to sort them immediately and he can resume shopping instantaneously.

If the user is unsure of an item's category, un-identified items are presented at the head of the Shopping Mode list to allow the user to keep an eye out for them and categorize them upon acquisition (thus categorizing the item for future trips). Items can have multiple categories that are implemented only in specific templates, allowing for inconsistently located items to be sorted according to shopping

experience (ie: 'Sinbad' assigns the "Specialty" sub-category to Capers in the Cambridge template, but they are "Aisle 2" in the Joe Trade Template).

Example of user interaction in the ESLA MVP:

User 'Sinbad' creates a New List based on a recipe he wants to make:

Cashews, cornstarch, (2) garlic, (3 c) basil, (1 c) pepitas, nutritional yeast, lemons, extra firm tofu, spinach, lasagna noodles

He then uses the Category interface to classify each item (in future iterations of ESLA, the categories and items are learned from each session, allowing the user to type less each time):

Bulk- cashews, pepitas

Grocery- cornstarch, nutritional yeast, lasagna noodles

Dairy- tofu

Produce- garlic, basil, spinach, lemons

Sinbad decides he is going to the Food Whole in Cambridge and selects the Template he has created for it. The list is reorganized by the store flow of the Cambridge template (Bulk, Produce, Dairy, Grocery), and 'Sinbad' enters Shopping Mode to allow for easy viewing of items while implementing an "item acquired" method (if an item is unavailable, it can be flagged and automatically added to a new list once the current one is completed).

But when he gets to the store, he discovers that there is a temporary encumbrance in the Bulk department! No worries as ESLA can easily be reversed while in Shopping Mode, allowing the user to invert their shopping list to accommodate disturbances in the flow. 'Sinbad' goes through the grocery aisles first, grabs his tofu, and then reverts the list to retrieve his bulk and produce items.

The result? Sinbad is able to get all the items on his list quickly and efficiently, and is able to review his list a final time to ensure completion.