Find It : *Painless Shopping*

Contents

By Moe and Sam

*1.0 Project Description / Context*

*1.1 Purpose*

*1.2 Motivations*

*1.3 Projected Client(s)*

*2.0 Overview of Product and Functionality*

*2.1 General Overview*

*2.2 Projected Functionalities*

*2.3 User Interface / Design*

*2.4 Project Resources*

*2.5 Software Category*

*3.0 Software Specific Requirements (Client Approved)*

*4.0 Target Audience*

*4.1 Uses Among Key Audiences*

*4.2 Targeted Demographics*

*5.0 Platform*

*5.1 Compatibility*

*6.0 Competition*

*6.1 Field Analysis*

*6.2 Find It vs. Competitors*

*7.0 Closing Remarks and UI Prototypes*

*1.0 Project Description / Context*

*1.1 Purpose*

To facilitate shopping by providing consumers with a list of local items based on a search query. (ex. Someone searches for khakis, they would get a list of khakis in Ottawa sorted by price, distance, etc.)

*1.2 Motivations*

The beginning of brainstorming for a product always begins with identifying communal needs. Find It was an app with admittedly many competitors. But we found that there were many apps that, for example, scanned barcodes to compare prices or showed you upcoming discounts, but there was no app that would allow you to search for an item and receive a list of options based on price, distance, and brand.

*1.3 Projected Client(s)*

For the purpose of this project, we are leaning more towards a client who shops very frequently and for a purpose. Really anybody that shops locally for items that need to be refilled would be a great target for us whether the good be clothing, art supplies, groceries, school supplies, etc. With this broad set of criteria, we do not limit ourselves to the sort of client that we can work with. Although a few people that we do have in mind are ;

* Mrs. Noel (In charge of buying art supplies on a budget, and since there are many art students this year she will be running out a lot more often)
* Mrs. Dreef / Bortnowski (They buy supplies for the math office at a teacher discount to sell back to students at a discounted price. You can never go wrong with more calculators and rulers)

*2.0 Overview of Product and Functionality*

*2.1 General Overview*

The app will provide users with an easy to use interface that will allow them to search stores in a set radius for items of interest.

*2.2 Projected Functionalities (\*\*Extended functionalities)*

* Search local stores
* Item prices
* Item stock
* List sort (to organize list based on price, distance, etc.) to fit a customer’s needs
* Store categories (ex. Browse clothing, technology, appliances)
* Which stores are closest
* Shopping list (save items for later viewing)
* \*\*Suggested items (based on previous searches)
* \*\*Set a budget(Gives different options to stay under budget) also includes setting a shopping list budget
* \*\*Tells you which stores have student discounts
* \*\*Input Different Coupons / Discounts you have and the prices will be applied

*2.3 User Interface / Design*

Our aim with Find It is to make it so that the user will open up the app and be able to find everything that they need to find and do what they need to do with no explanation.

A good user interface is one that is intricate and thoughtful yet easy to look at and not jarring. Achieving such a feat is worthy of praise in the design community, and that is what we intend to do.

In terms of design, we will base the theme of the app off of material design with an autumn color palette and polygonal texture base. (If we have time, we’ll make different color schemes so that the user can choose between them)

Refer to Figures 1 and 2 in section 7.0 for samples of the UI that we will be integrating, or at least the style of design we’ll be using.

*2.4 Project Resources*

Our project will make use of google maps as a software engine, relying on its functionalities to return a list of a type of store in an inputted radius. Fortunately, there are various commands that can be inputted into google to make use of this system.

As far as design goes, there is a plethora of materially designed apps on the google play store, so there’s no shortage of inspiration there. Also, we will have access to the full Adobe Creative Cloud throughout the entirety of the project. That is huge for us because it makes the design process about 10x less painful.

\*Side-Note : Having access to a Shopify member throughout the duration of the project will be a huge asset and will enrich the learning process even more.

*2.5 Software Category*

This app classifies as a Shopping application.

*3.0 Software Specific Requirements (Client Approved)*

Facilitation of local shopping for users. This app will allow users to look up the items they desire to purchase before they even leave the house, letting them judge the budget and plan not to overspend. Though this is the main idea driving the program, one can refer to section 2.2 for specific functionalities that need to be working to hit the requirement.

*4.0 Target Audience*

*4.1 Uses Among Key Audiences*

Key audiences will be cautious and casual shoppers that have a desire to see what items are available locally and what prices. For the cautious shopper this will allow them to really get the best price for the item they are looking to buy. For the casual shopper this will allow them to browse items of interest in the local area.

*4.2 Targeted Demographics*

The targeted demographic is anybody that owns a smartphone and goes shopping. We can market the app to this group as something that will allow them to browse local stores without leaving the house, giving them the opportunity to establish a budget.

*5.0 Platform*

*5.1 Compatibility*

As both Sam and I own Android phones, we will work to develop Find It for Android first. And at later stages, we will translate into iOS and maybe even desktop. This way, we can optimize QA testing as we both have access to a testing platform.

Also, the development in Android is beneficial to our case as it provides a common ground for us as developers in its official development language being Java. We are both proficient in Java, so starting with Android will result in the fastest high-quality completion of the project.

*6.0 Competition*

*6.1 Field Analysis*

In doing research on what the market already had in terms of shopping facilitation apps, and we came across three big ones.

RedLaser (iOS, Android, Windows Phone) - This is an app that allows you to scan barcodes in a store and get an instantaneous price comparison. (Requires data) A plus to this app is that it also stores loyalty cards and its system allows it to include both online and retail stores.

ShopSavvy (iOS, Android, Windows Phone) - ShopSavvy is an app that combines the functionalities of RedLaser with all that you would need to know in the realm of savings and discounts to make the best decision on what to buy. You can simply input an item that you want and it will keep an eye and notify you when it is on sale at a store. It will also tell you if something is on sale somewhere else if you scan the barcode in-store. (This feature requires data)

ShopAdvisor (iOS, Android, Windows Phone) - ShopAdvisor combines both the functionalities of ShopSavvy and RedLaser as well as allowing you to scan items from magazines. It will also watch the price changes (peaks and crashes) to let you know when an item becomes the price that you are willing to pay for it.

*6.2 Find It vs. Competitors*

For the sake of repetitiveness I will refer to the previously mentioned competitors as ‘the market’. Also, note that most points made in this section refer to the ‘ShopAdvisor’ app unless specifically stated otherwise because it combines the functionalities of the two above.

If we list side-by-side the functionalities of ShopAdvisor and Find It, we get a comparison such as so:

|  |  |
| --- | --- |
| Find It | ShopAdvisor |
| * Search local stores * Item prices * Item stock * List sort (to organize list based on price, distance, etc.) to fit a customer’s needs * Store categories (ex. Browse clothing, technology, appliances) * Which stores are closest * Shopping list (save items for later viewing) * \*\*Suggested items (based on previous searches) * \*\*Set a budget(Gives different options to stay under budget) also includes setting a shopping list budget * \*\*Tells you which stores have student discounts * \*\*Input Different Coupons / Discounts you have and the prices will be applied | * Item prices * Price changes (Watches as the price rises and crashes and lets you know when it reaches the price that you are willing to pay for it) * Search local stores * Scan items for instantaneous price comparison (Scan a watch in-store and it will pull up a list of other stores that sell the same product for cheaper) * Scan magazine items to see which stores sell them * Recommendation in which you sign up to an email list / social media and it sends you (usually sponsored) items that are going on sale that you might be interested in (This one of the ways that the free app generates revenue |

I’ve highlighted all of the unique functionalities on either side of the table. As you can see, there is a lot of deviation from the current market with Find It. Some of the major differences in functionality include;

* Find It’s budgeting function which allows a user to create a shopping list and a budget which then tailors all search results to allow the user to remain under budget (and provides options in some cases to save money). Now, this functionality runs into issues when a user types in something unreasonable, like a wedding ring for $5. There will be a system set in place to prevent that from messing with the app.
* Find It’s discount function which allows a user to input coupon codes into a personal ‘coupon book’ that is saved to the user. The coupon must contain information such as expiry date, applicable stores, and items included. All search queries will then be tested to see if the coupons are applicable and then adjust prices.
* Also under the discount function, is the special user status function that allows a user to input any information about themselves that may make them eligible for discounts at various locations. (For example, Under Armour at Tanger Outlets offers a 25% discount on all clothing items to anyone who’s been in the army) This may include age, occupation, student status (full-time / part-time), etc.

In doing this research, we have found that ShopAdvisor has more of a focus on in-store price comparison or consumers that know *exactly* what they want to buy. (Just not when or where) Whereas Find It places more of an emphasis on browsing. What I mean by that is ShopAdvisor requires specific brands and information about a product in order to maximize the efficiency of results. But since Find It will only work in Ottawa, we have the freedom to allow more broad search queries. For example, on ShopAdvisor, if you were to type in ‘pants’, you would get a whole slew of results that are really not what you are looking for. So you would instead have to type in Slim-Straight Jeans, for example to narrow the results. While on Find It, you could type in pants and since the app only works locally, it would work based off of previous search queries to determine what type of pants you would like. (E.g. A user who prefers tight clothing usually won’t be looking for baggy wind runners) This prevents users from being bombarded by results, and the best part is that they can change whether or not they want Find It to work off of recommendation or not.

*7.0 Closing Remarks and UI Prototypes*

Throughout the entirety of this concept document, we have discussed the pros and upsides to Find It. But in the closing remarks, we will play devil’s advocates and discuss some of the issues and problems with our design plan.

Firstly, it will be very difficult to implement an algorithm that will search EVERY store in Ottawa the same way every website has its own design and layout. This will mean that we will have to implement (almost) every store that the app will include. Apart from the framework and the design, that will be the core workload throughout the duration of the project.

Secondly, since that approach will heavily impact the efficiency we will have to very intricately map out our strategy in the project plan and spend a lot of time on developing algorithms before we even begin to code. This will most likely take up the first week of development. The upside to that is that after we set all the framework, it should just be a matter of filling in the holes. But since we know that’s never how it works in the SDLC, we will adopt a flexible SDLC model that will allow for tweaks and fixes along the way in case we encounter a pitfall.

We will conclude our Concept Plan with several (rough) prototypes of the sleek UI for Find It. This is kind of mandatory for me as a graphic designer as I’m literally obsessed with material design.

|  |
| --- |
| Find It - Prototype 1.png |
| Figure 1 |

|  |
| --- |
|  |
| Figure 2 |