

# AllSortz App Spec

*Last updated 8/7/2012 by Matt Zoufaly*

## A.) Displaying Businesses

### 1. Overall Approach

The goal of the listings page is to always show users an ordering of businesses. AllSortz, in keeping with its name, calls these orderings “Sorts”. The Sorts can be ordered anyway the user would like. By default, AllSortz will display enough information to give the users a sense of what a business might be like (e.g. what kind of food they have), and some sense of the businesses’ “quality” (e.g. its rating)

### 2. Location Values

Location is a critical issue in dealing with brick & mortar businesses to ensure relevancy. In particular, a large search radius might display results that are humorously far away (e.g. San Francisco results when searching in NYC) or cut out results unnecessarily. The radius can be different for each user, depending on access to public transit, cars, and other transit.

#### i. Defaults

By default, we use your GPS location / best approximation of where you are. The default radius in which we’ll display businesses will be 2 miles, since this is a reasonable walking distance. However, you can configure the radius to be larger or smaller in your account settings. 2 miles isn’t a strict cutoff, but is a boundary at which a business had better be “very appealing” to listed in a Sort.

#### ii. Modifying Location

Location is continually updated based on your GPS location / our best approximation. However, we will allow users to set their “location” by using the map view. This is important so that a user on their way to a city / destination can still check to see what’s happening in that area.

- A button on the map-page that drops a pin will be sufficient

#### iii. Modifying Sort Radius

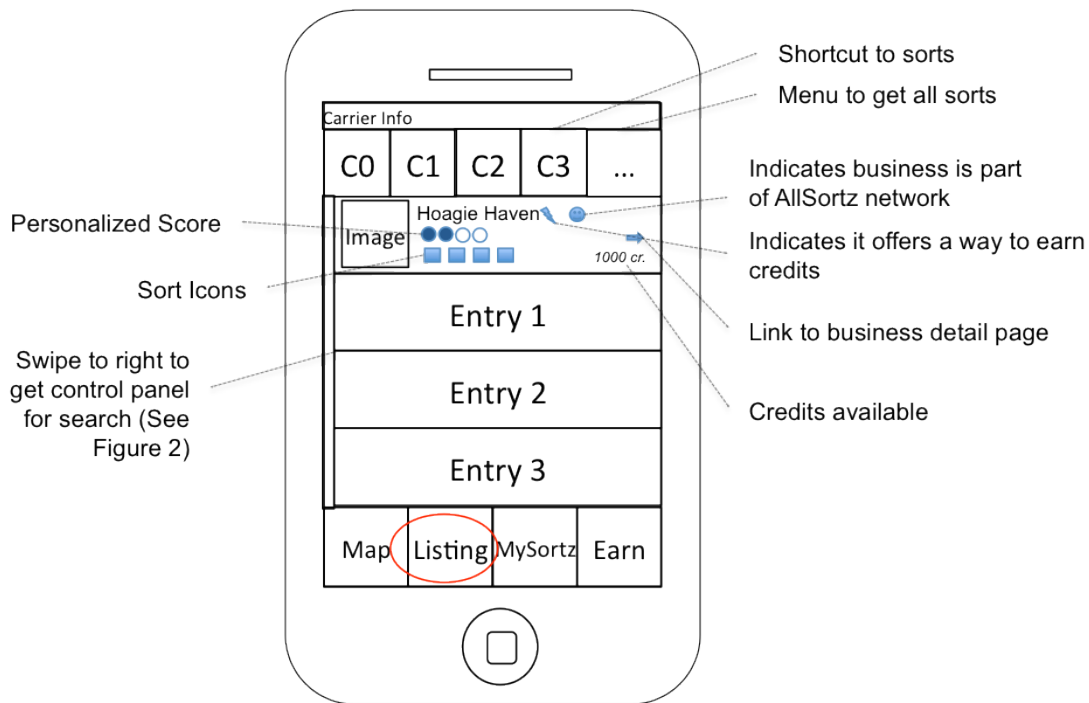
A person’s apprehension / likelihood of traveling far is going to remain relatively constant over the time they’re using the app. Actual distance numbers shouldn’t factor into a typical Sort. Thus, the radius for Sorts can be configured in a user’s Account Settings page. A potential list of options might be: 0.5 miles, 1 mile (short walks), 2 miles(longer walk), 5 miles (bike ride), 10 miles (car ride), 25 miles(long car ride)

## 3. Listing Page Design

Currently, our sketch for the business listings page (8/2/2012):

# Business Listing Page

(seen after tapping 'Listings')



**Figure 1**

## i. Business Listing

The focal point is the list of businesses in the center of the screen. Each listing consist of:

- A picture
  - o There should be a default picture for each business.
- The business name and its general location (neighborhood, town, etc.)
- What sort-categories it is under (or at least the top ~6)
  - o Tapping the sort icons should bring up a popup that gives at least the text for each sort, as well as the businesses' rating for that sort
- The score our engine has determined you would rate it as
  - o Tapping your score brings you to an info page that might bring up a popup that explains why we think this is a good score for you. It should also allow you to edit your scoring
- Whether or not the business is having a deal
  - o In the future, might include "types of deal"
- Whether or not the businesses is part of our network
- Credits available to buy coupons, discounts, etc.
- A link to the detail page for the business

## ii. Additional Page Functionality

- Swiping from left to the right of the screen will bring up the Search Control Menu page (described in the next section)
- The icons on the top are selected by the user, though we will have default searches when the user first joins the app (Discussed in the Sorting section)
- Pulling down on the listing reveals a general search bar.
  - o Can be used as a normal search bar would in Google.

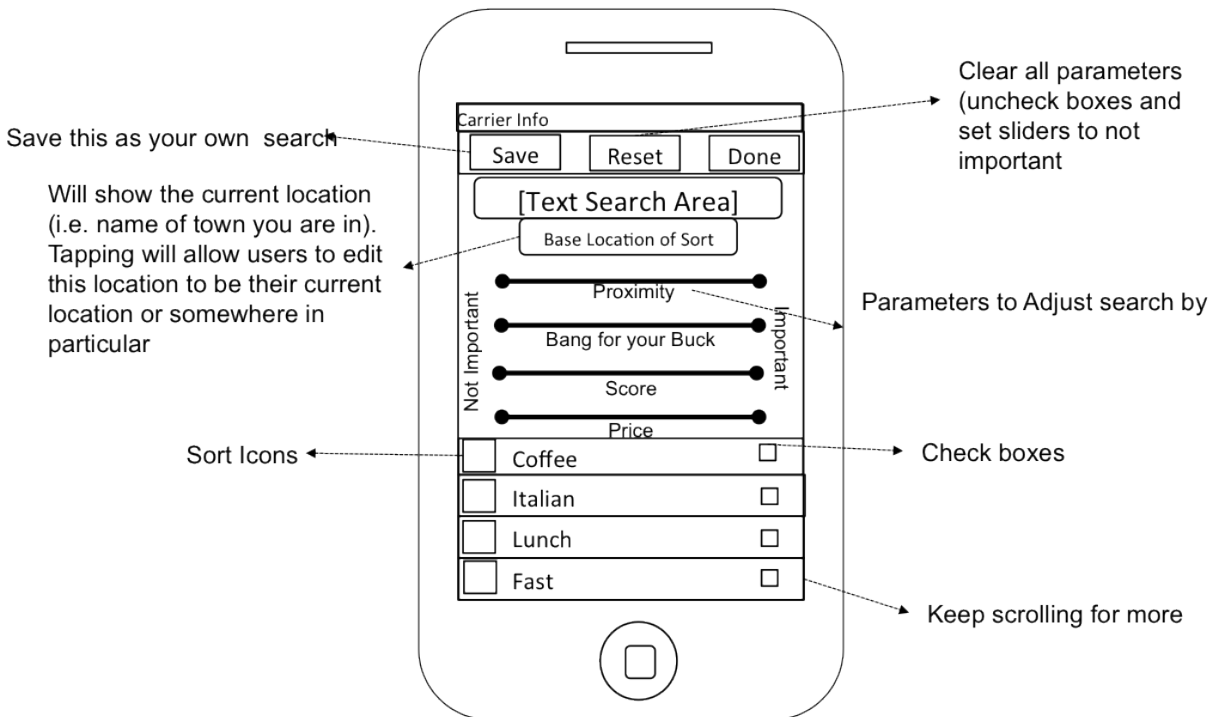
- This feature isn't necessarily critical to the app, which is why it is hidden away as an Easter-egg type feature.

## 4. Sorting Control Menu Design

This page is opened when the user swipes from the left side of the Listing page.

### Sorting Control Menu

(from Figure 1)



**Figure 2**

The page is set to whatever the values of the current search are. It might contain the parameters for our default searches, or it might be the parameters from a friend's search. Users can tweak the search parameters here.

#### i. Text Search Option

The text search here allows for users to include some kind of text in their Sorting Control. For example, they might want to search for text that isn't really part of a sort category such as "Winter Wonderland"

#### ii. Bars at top

The bars at the top of this screen aren't cutoffs: they are weights. If they are all set to the right-hand side of the phone, then they are considered important (and vice versa). Current options:

- Important of Price
  - Is it cheap food
- "Bang for your buck"
  - What do you rate the "value" of this businesses food
- Importance of distance

- Is it important to be down the block, or are you willing to travel?
  - Actual distance cutoffs can be set for your account
- Importance of Score
  - Do you want a high-scoring restaurant, or do you care about other stuff only

Having score set to the max, while the others are set to 0, means that we're going to find the place that has the best score regardless of its distance (although there will always be some global default cutoff so that San Francisco results are not returned in NYC). Setting all of them to max is also fine. This ends up finding the "best place, with the best value, with the best score, closest to me". The settings change the ordering proportionally, thus having them all at max is the same as having none set.

### iii. Sort Categories

The category searches allow you to find something in particular that you're looking for. Not setting anything is the same as saying you have no preference.

- There needs to be a means of checking all categories, clearing all, etc.

### iv. Filtering

There is some limited filtering capability

- Different than sorting since businesses will *NOT* show up if conditions are not met
- Filter if there is a deal available
- Filter if the business is in our system
- Filter if business is currently open

### v. Saving

- Tapping save (on the top-left) will allow you to save this particular configuration to be used to sort later (see sorting in next section). The sort will show up in "Your sorts"

### vi. Defaults

- Whatever the first sort is will function as the default sort. If this is the first time the user is using the app, finding all top-rated businesses will be the default (since it will be the first button)

### vi. Use Cases

- User maxes the "score" bar and checks "sandwiches" to search for high-scoring sandwiches within default distance parameters
- User wants to search for a particular tag word under the current parameters. The tag can be put into the text-search area
  - E.g. user wants to search for winter wonderland
  - Full-text search will be covered in more detail later

## 5. Custom Sorting Buttons

On the iPhone there is enough room for roughly 5 buttons on the top of the screen (if they are to be square). They are used for ways to sort businesses. For example, if a user has a "sorting" that finds delicious coffee in the area, there should be a button that does just that.

### i. Default Behavior

AllSortz maintains roughly 20 different kinds of Sortz that we think people would like to use (e.g.):

- Top Rated Dinner (anywhere)
- Top Rated Coffee (near you)
- Best value food (anywhere)
- Best bars (near you)
- Best Brunch (near you)

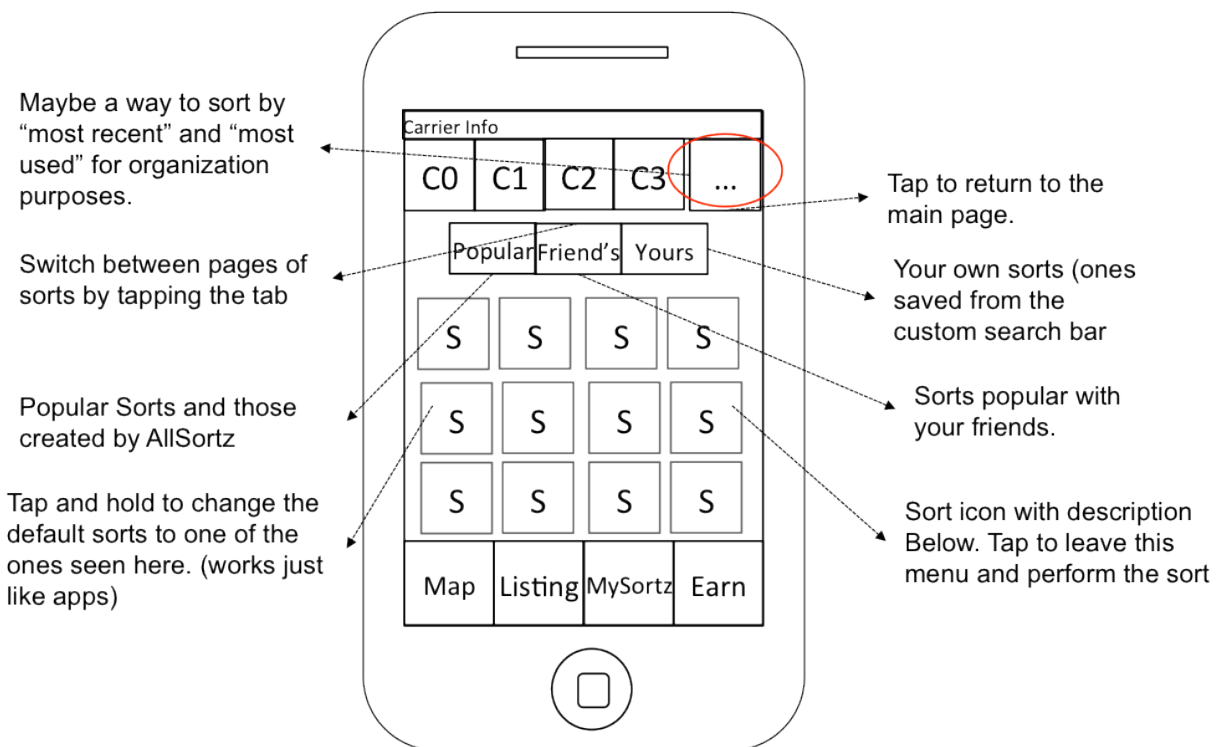
Each of these will have corresponding settings in the “Sort Control Menu” that brings up the results.

### ii. Customizing Sorts

Once the user taps ‘...’, a custom sort menu is brought up:

## Custom Sort Menu

(From tapping ‘...’)



**Figure 3**

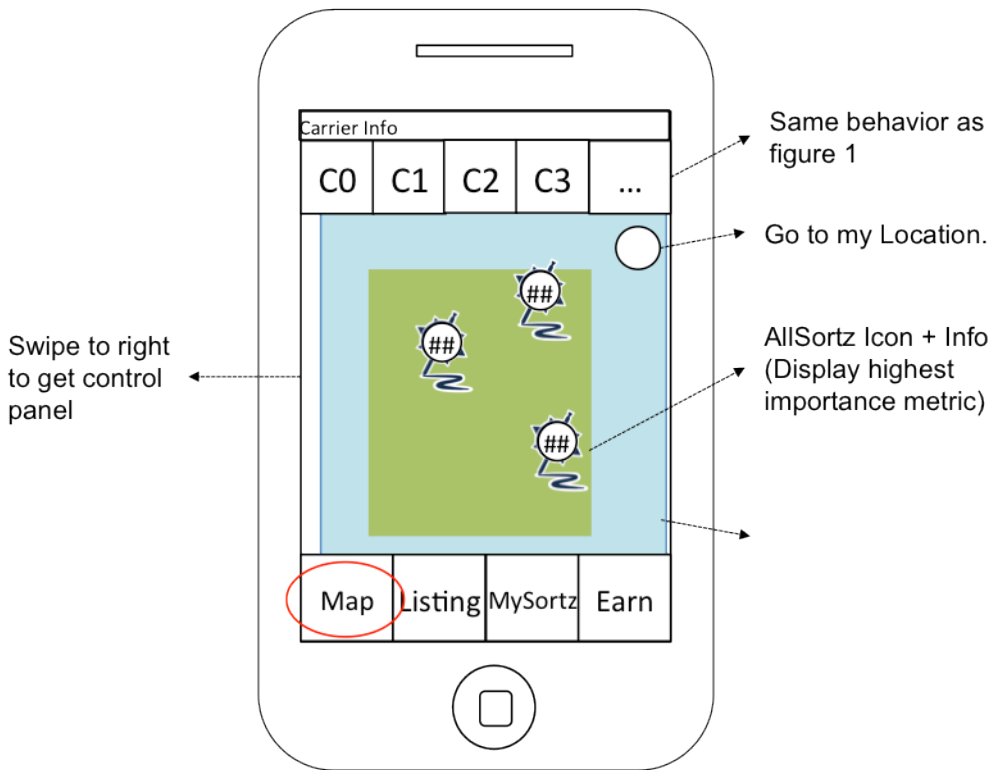
Custom sort menu

## 6. Map View

Similar to the listing, except it excludes any kind of distance metric (either the limits defined in the user’s account page, or an importance rating).

## Map View

(From tapping Map)



**Figure 4**

### i. Default Behavior

- Center on where the user is, and show a 2 mile radius around the user.
- Whatever sort is currently applied from the listings page should be applied to maps (excluding any distance parameters)
- If scrolling stops momentarily (~2 seconds), redo the search in the region the map is current viewing

### ii. Icon

- The icon will be the AllSortz icon, except the center of the icon will contain one of the following (depending on the particular sort)
  - o Value metric
  - o Score
  - o Avg Price per meal

### ii. Future Ideas

**TODO:** draw on map with finger to get a region to search in

## 7. Business Details

A more detailed version of the listing.

## Detail View (1)

(Business details)

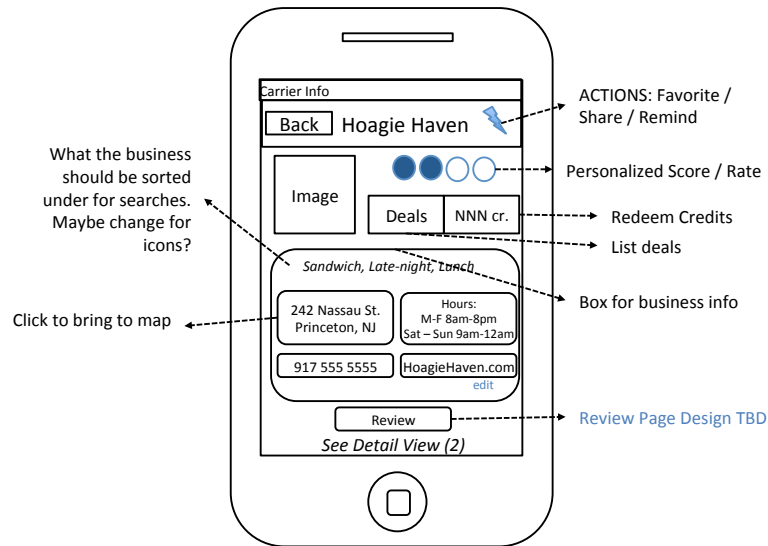


Figure 6a

## Detail View (2)

(Business details)

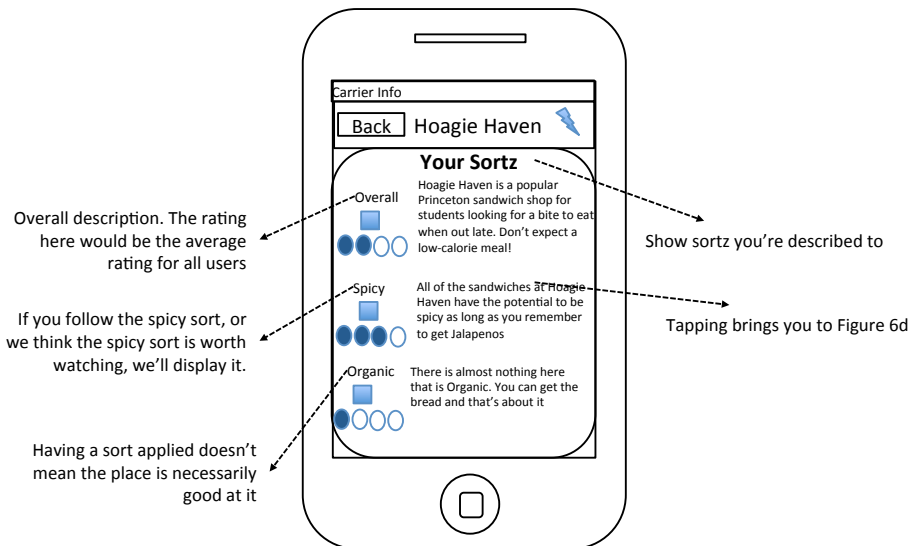
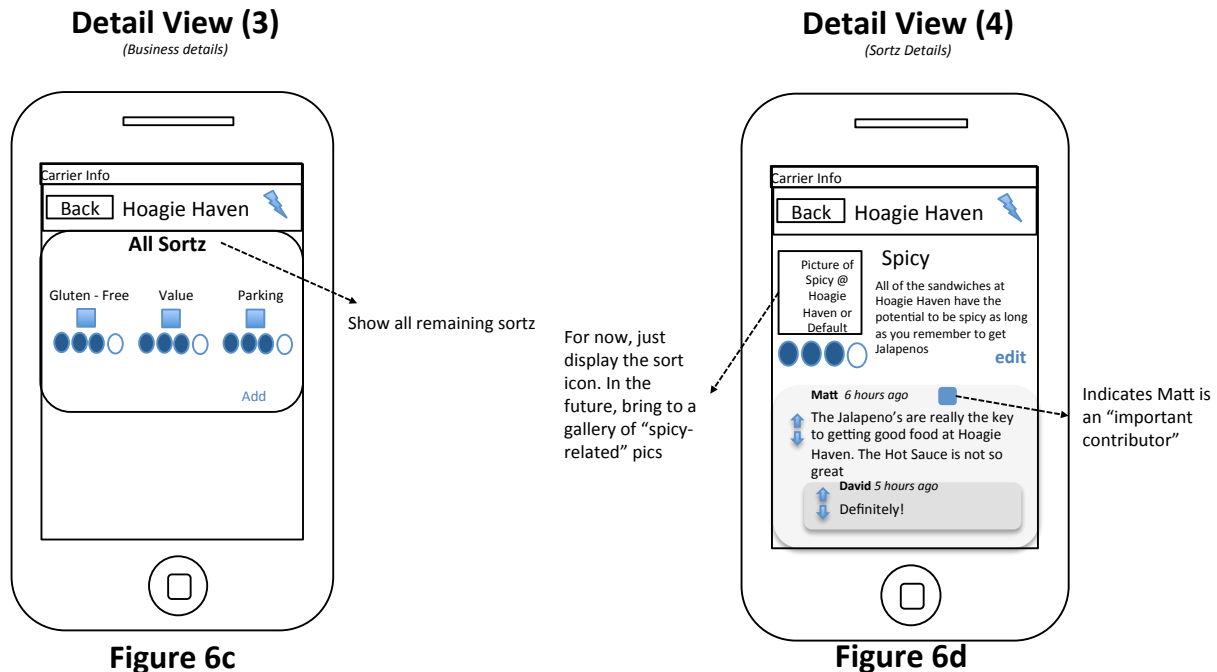


Figure 6b



**Figures 6a-d**

**The Deals / Redeem page is in Figure 7 and 8**

### **i. Overview**

This is where the user will get their *customized* review for a business, as well as other useful information about the business, such as its relevance to various sorts, pictures, and a general discussion about the business.

### **ii. Business Sort Reviews**

- There will always be the initial three-four sentence moniker associated with a business. This gives a description of the business for anyone who is viewing the page, regardless of interests. It should be thought of as the most objective way of classifying a business.
- There are also the reviews generated from the sorts. We organize these reviews by showing ONLY what sorts we think you’re interested in
  - **TODO: determining how people are interested in sorts. Is it still manually populated, or is there an automated way of doing this?**

### **iii. Discussions**

- Threading & Voting
  - Discussions should be threaded, and have up-vote and down-vote functionality (mimic reddit.com’s implementation of comments).
  - Upvoted comments trickle towards the top of whatever thread they are in, and the “root-level” threads are organized by a mixture of the time they are posted and the amount they have been updated.
    - For the interim, just use the number of votes.



- Types of discussions
  - o The **business discussion** is meant as kind of a “catch-all” for people who just want to leave a comment about the business and are not interested in posting under a particular sort
    - E.g. I walk in at a busy dinner hour, and leave a comment about how well the business is handling the crowds of customers (or not)
  - o The **sort discussions** are more targeted at resolving the synopsis review for the sort. Thus, the discussions are similar to the “discussion” page on Wikipedia where users collaborate and talk about how the article should look
  - o The **photo discussions** are targeted to attract people who are interested in Pinterest-style applications (where images are the primary means of displaying information). In order to appeal to the widest user base, AllSortz allows users to also just post pictures and have discussions about pictures (no sorts, no reviews, etc.)
    - **TODO: have a mode of AllSortz where the ONLY thing displayed are business images and the review.**

#### iv. Deals / Ways to get credits

Figure 7 and 8 show the basic functionality to get credits and then redeem them for deals. Essentially users do things to get ‘credits’ which can be used as gifts or to buy deals.

##### Earn Credits

(tap earn credits on the business details page)

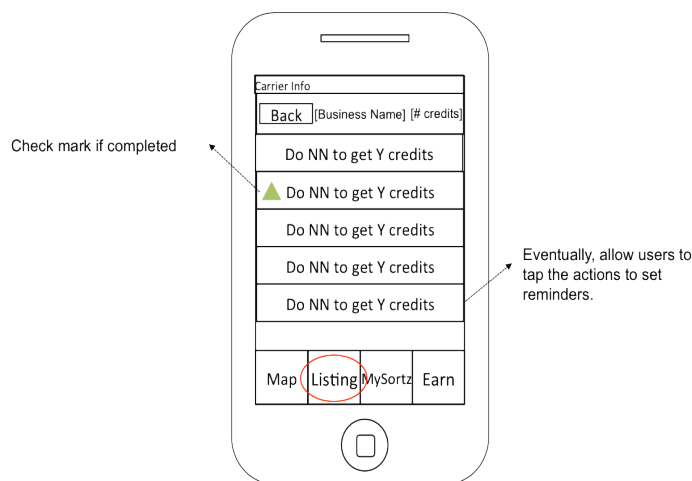


Figure 7

##### Redeem Credits

(tap redeem credits on the business details page)

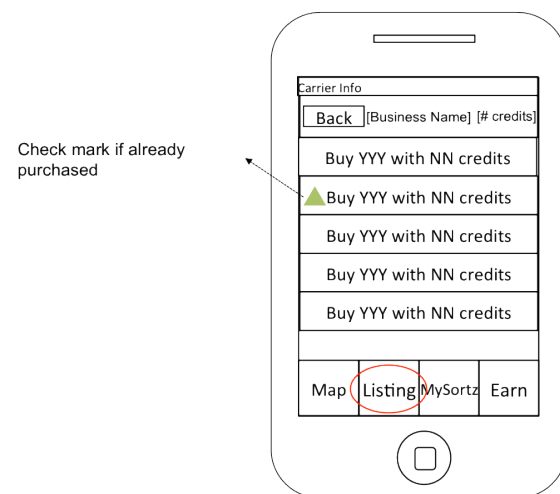


Figure 8

- See how many “coins” they have a business (from checking in, buying stuff, etc.)
- See what businesses are offering to buy for those “credits”
- See how they can get more points
- Allow the gifting of points to others. For example, I can give my 1000 Starbucks points to my friend as a Thank You.

Allows businesses to:

- Offer users an incentive to perform an action (e.g. buy something, or check in) for a certain number of “coins”
- Offer deals that can be bought with those coins

## B. User Accounts

### 1. MySortz Tab

Described in Figures 9 - 12

#### MySortz Feed

(tap MySortz and tap Feed)



Figure 9

#### MySortz Friends

(tap MySortz and tap Friends)

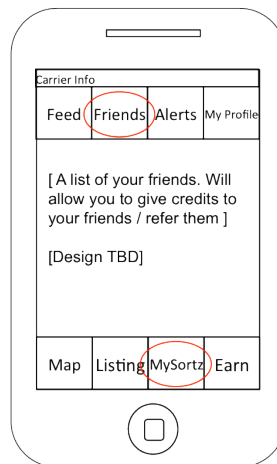


Figure 10

#### MySortz Alerts

(tap MySortz and tap Friends)

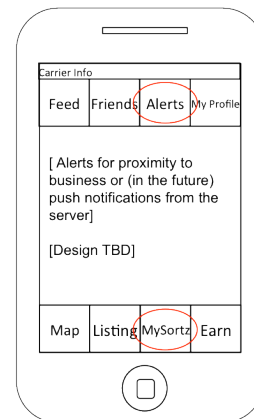


Figure 11

#### MySortz Profile

(tap MySortz and tap My Profile)

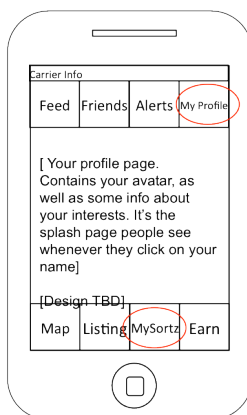


Figure 12

The Feed tab allows users to

- Check a feed of things happening at businesses
- See what their friends have been up to

The Friends tab allows users to:

- Search for friends on AllSortz
- Delete friends
- Give friends gifts / coins

The Alerts tab:

- Placeholder for businesses to provide sponsored push-advertising
- A way to receive alerts for new deals / ways to earn coins

Profile tab:

- Contains your profile that you see when users look at your page
- Allows for basic settings as well (e.g. Distance preference)

## C. Scanning for Payment

## D. Rating System

### 1. Overview

#### i. Intro

The most complex component of AllSortz. The overall approach is to gather meta-data about users, such as what they like doing and the food they like eating, and then observing how businesses are categorized and rated. With that information, AllSortz will provide recommendations tailored for each user.

#### ii. Scale of ratings.

Determining the scale for the ratings is important insofar as the user experience, but from a usability standpoint, it doesn't make a huge difference. Using a 2 star (like dislike) system will likely not yield drastically different results than a 5 star system would. As discussed in a meeting on 7/23/2012, it might even be good to allow users to customize what each "star" on a 5-star rating means.

Right now, AllSortz uses 4 stars (hate, meh, like, love) for business ratings and up/down votes for comments and for the ratings of "sort categories" in businesses. This is to minimize on complexity and encourage users to "click around" rather than deal with dropdowns. Implementation wise, it is easy to change since the ratings are just integers in the database.

### 2. Sort Categories

#### i. Categories

AllSortz maintains a list of "sort categories". These are essentially descriptors for restaurants. They can be general such as:

- Lunch
- Brunch
- Dinner
- Fast

Or they can be more specific and discuss types of food

- Brazilian
- Italian
- French
- Coffee

- Dessert

And finally, they can be a descriptor for people with disabilities / food limitations or other miscellaneous concerns

- Kosher
- Vegan
- Gluten-Free
- Vegetarian
- Handicap Accessibility
- Family-Friendly
- Parking

## ii. Application and Rating

Businesses can have these descriptors applied to them, and then the descriptor can be weighted negatively or positively. Thus, a sort is either *applied* to a business or N/A. Any user can apply a sort as they see fit. Once applied, a sort is rated. This rating indicates “how good” this sort category is handled at the restaurant.

A sort can be applied even if it is technically N/A. A business that has received 100% negative ratings for a sort is sorted the same as a business that does NOT have the sort applied. Thus, users can add sorts just to complain about a lack of something without disrupting the sorting capabilities of AllSortz

## iii. Examples

A sandwich shop that does not sell coffee

- A user comes along and applies: Spicy and Sandwich.
- Sandwich is up-voted because the place generally has good sandwiches
- Spicy is down-voted because the spicy sandwiches are pretty bad
- Coffee is excluded because the business does not sell coffee,
- Another user comes along and applies coffee, with a negative rating asking why coffee is not on the menu
- The business leaves a “business response” saying they’re not going to include coffee on the menu

# 3. User Data

## i. Traits

When a user makes an account with AllSortz, we optionally allow them to populate their make interests manually. For example, you can indicate that you like spicy food, or you like going out for dinner. These “traits” are not used explicitly when you’re searching for items, but are data that we collect so we can see how users think of themselves as they give ratings

## ii. User Sorts

Users can follow particular “Sort Categories” as well. Following a sort category just says that the information associated with the sort is stuff that is typically relevant to you. Thus, we will display this information by default when you look at a business. This also allows us to do simple recommendations. For example, if a user is following “Coffee” then whenever a new coffee place is listed, we know it might be something that we recommend to you.

**TODO:** We can go a step further and even figure out your relative interests based on places you’ve rated highly, thus those Sorts are automatically added

## iii. Combination?

It might be the case that traits and user sorts are redundant. However, sticking with the methodology of collecting more data than we can synthesize indicates we should keep these two features separate for now. In any event, they are both optional and do not hinder usage of the application

## 4. Recommendation Engine

### i. Goal

Given all of a user's ratings, and all of a business's ratings, figure out what the user's rating would be for a business that they have not been to. This is a key feature of AllSortz, and will be a large factor in our position over foursquare and Yelp. There are several approaches to recommending places

### ii. Full Recommendations using MF

Using the existing matrix factorization technique in the repo, it's possible to simply return a number to the user for the businesses they have not been to. However, this is difficult to do on absurdly sparse tables (as it would be for the first couple of months).

- Pros
  - Black-boxed. The algorithm is already implemented and is hooked up to run.
  - Allows for complex relationships amongst businesses and users
    - E.g. a user A has factor Y as does user B. If user B and C also share interests, then factor Y can be associated with user C
- Cons
  - Difficult to use on sparse matrices, and is computationally expensive on large matrices
  - Difficult to determine *how* we came about with the recommendation
    - e.g. saying "Based on your interest in spicy-food, we recommendation ..." or "we think you'll score this  $\frac{3}{4}$  based on your interest in Hoagie Haven"

### iii. Aggregate of Ratings

We can also use the average ratings in the sorts that are relevant to you to figure out the "average for you".

- For example, if you're interested in spicy, Italian, and dessert, then AllSortz would guess that for a place that has 4/4 for each of these ratings (and  $\frac{1}{4}$  for everything else) you'd rate it with a 4/4. NOTE: this is essentially the same as the matrix factorization approach, but is being done manually by use of the "Sort ratings"
- Pros
  - Much more transparent to the user how the ratings were obtained
    - E.g. "Based on your interest in spicy food, we recommend..."
- Cons
  - Performs the same essential computation as MF, except the factors are inserted manually instead of automatically
  - Will still fail in the sparse case since an unpopulated business will not have sufficient data to give a recommendation

### iv. Combination of Approaches

We can allow recommendations for the overall business to be driven by MF, while the ratings on the "sort categories" inside the business are driven by simply averages. Thus,

Matrix Factorization is utilized for its accuracy. Also, the recommendation engine works even if the “Sorts” under a business are incomplete, or there are similar sorts (which would throw off the weighting otherwise).

## E. API

### 1. Overview

AllSortz uses a RESTful API that uses a Django web framework running on an EC2 instance. Querying the API will return a JSON object with the results. The API uses a Postgresql database backend that is identical to the data seen on the website. Thus, using the website is no different than interfacing through the app. This section will describe how the API is laid out, and how it is queried.

[SEE docs.allsortz.com \(API Spec\)](https://docs.allsortz.com) for up-to-date API information

### 2. Functions

#### i. Search businesses

Endpoint: /businesses

Parameters:

- Weights
- Categories
- User's location
- Filters
- Text search

Returns: List of results

#### ii. Business

Endpoint: /business

Parameters:

- Business ID

Returns: Details for specific business

#### iii. Business Category

Endpoint : /business/category

Parameters:

- Category ID

Returns: Details for category in a business

#### **iv. Comment**

Endpoint: /comment

Parameters:

- ID or list of IDs

Returns: List of comments

#### **v . Comment search**

TBD

#### **vi. Photo**

Endpoint: /photo

Parameters:

- Photo ID or IDs

Returns: Information about the photo

#### **vii. Search photos**

Endpoint: /photos

Parameters:

- Ordering (by date or by rating)
- Range (1, 1-100, whatever)
- Type (business, user, etc.)
- ID for type

Returns: List of photo information

#### **viii. (Custom) Query**

Endpoint: /query

Parameters

- Search ID

Returns: Whatever

#### **iv. Search queries**

Endpoint: /queries

Parameters:

- Ordering
- Range
- Type

Returns: List of queries

#### **v. Post Rating**

Rate business, photo, category within business, comments, possibly more

Endpoint: /{entity to rate}/rate

Entity can be:

- /business: the overall rating of a business
- /business/category: the rating of a category within a business
- /comment: the rating of a comment
- /photo: the rating of a photo

Parameters

- Rating

## **F. Competition**

### **1. Reviews**

- Yelp
- Zagat / Google
  - o “Top Reviews”
- Foursquare
  - o Allows you to “check-in”, leave tips, etc.
- CitySearch
  - o Yelp-like search parameters
- AroundMe
  - o Uses Foursquare API

### **2. Deal Finders**

- Living Social
- Groupon
- Promoverse

### **3. Mobile payment / Use app to get something**

- Starbucks payment system
- LevelUp Payment System
- Stamp
- Foursquare
  - o Will give you money back on a deal if you use Amex
  - o I haven’t been able to do it...

### **4. Checkin Capabilities**

- SCVGNR (Scavenger)
  - o Allows you to setup “Things” your friends can do
- Foursquare
- 

## **G. Meetings**

### **1. 7/23/2012**

- Development of the manual is key. Do not slack on it
- Need to have a functional version of the app for VC meetings, even if there are only a couple of businesses
- What are the default values?
  - o Default search for listings? Default map location? Default preferences for users?
- Non-logged in users on iPhone OK.
  - o Only ‘earn’ required additional user information
- Higher levels of categorization for the future
  - o Amusement parks
  - o Salons?
  - o Etc.



- Reputation of a business?
  - o Is it a tourist location?
- References to other establishments should be OK
  - o Refer to Small World when talking about “Local” in Starbucks
- What happens when you open the App?
- 

## H. Development Notes

### 1. TLS

- Use of TLS would be nice, and allsortz.com is currently ready to use a certificate, but the price is somewhat steep. See /ssl/setup.txt for information on how to setup SSL certificates

### 2. iPhone Authentication

- In order to keep development time down, we’re currently going to use a hybrid approach of Django’s built in authentication system and code that I’ve written on top of it
  - o AllsortzUser (defined in ios\_interface/models.py) contains a reference to the deviceId of a user. The device ID is enough to establish a user for now, so we generate a fake username, and password (generated\_password).
- A user that uses the API with a device that has been seen before is automatically authenticated
- **TODO: In the future, use OAUTH?**

### 3. Server Infrastructure

- The actual API is running on Django, on a MEDIUM EC2 instance.
- The database (Postgresql) is running on a Micro EC2 instance

### 4. Python Libraries

- Regular python libraries are fine for the most part.
- There are too many Django packages out there. I tried to use them initially, but they became too overwhelming (dealing with dependencies and learning other people’s code). Unless what you’re trying to do is incredibly complex and you have NO IDEA how to do it, roll your own Django code. I suspect if you have NO IDEA how to do something, you shouldn’t be doing the development to begin with
- Packages (both Django and Python) used so far
  - o django-registration
  - o django endless pages (to get the scrolling action seen on the front page)
  - o geopy, numpy, and scipy (distances, recommendations)
  - o

5. ...

6. ...

