# CrowdBuy - Group Ordering Made Easy

https://apps.facebook.com/509825915758193 *Members: Fazli Sapuan, Jerome Cheng, Joel Low, Vishnu Prem*
CrowdBuy is an app that helps you find friends who are interested in buying the same product as you are. Through user research we found that many of our friends in Singapore are reluctant to buy goods from overseas retailers because international shipping rates are exorbitantly high: occasionally you might end up paying 30-80% of the product's value in shipping costs alone. CrowdBuy aims to solve this by providing a platform for people to find other friends who might be interested in buying the same product as well. Using CrowdBuy, a user can post an item that he wishes to buy, advertise it on his timeline and try to get his friends interested in that product. Others who are interested can commit their purchase on CrowdBuy. Once a sale's expiration date has been met, the creator of the listing can then buy the product in bulk, saving on average shipping costs.

**Use case:**

- Joel wants to buy a YubiKey from the US. He realizes that it would costs $20 to ship this $20 product back to Singapore.
- Upon hearing about CrowdBuy, Joel posts in the app that he's planning to buy a YubiKey from the US within the next 2 weeks and if any of his friends are interested, they can commit to buy one too.
- After reading Joel's timeline post about his interested to ship a few YubiKeys, some of Joel's friends, such as Vishnu, Fazli & Jerome go into CrowdBuy and indicate their interest and quantities for purchase through Joel too.
- At the end of the 2 weeks, Joel checks out the number of orders through CrowdBuy and purchases them from his online retailer. As this is a bulk purchase, the shipping cost is distributed among his friends. Joel sends a notification through CrowdBuy to Vishnu, Jerome & Fazli that he has purchased X quantity of YubiKeys.
- Once the shipment has arrived, Joel can sent a notification through CrowdBuy to let his friends know that the shipment has arrived and he launches a Facebook group chat to arrange a time & place to meet and handover the items.

**Alternative use cases:**

- Fazli is travelling to the US next week on holiday. He plans to purchases a ChromeCast from Amazon once he's landed. As Fazli is a nice guy, he makes a new post on CrowdBuy to inform all his friends that he is planning on buying a ChromeCast in 2 weeks. He wants to see how many people would like to capitalise on his presence in the US to purchase other things.

## *Aspiration 1 - iFrame or Standalone app*

For an app of our purpose, we decided that it would be most useful as an iFrame app that lives inside Facebook. CrowdBuy would be more useful for you if more of your friends use it and hence we decided that it would make most sense living inside Facebook itself. Also, as we plan to make posts on the user's timeline, having the app live in an iFrame would it the most 'integrated' feel.

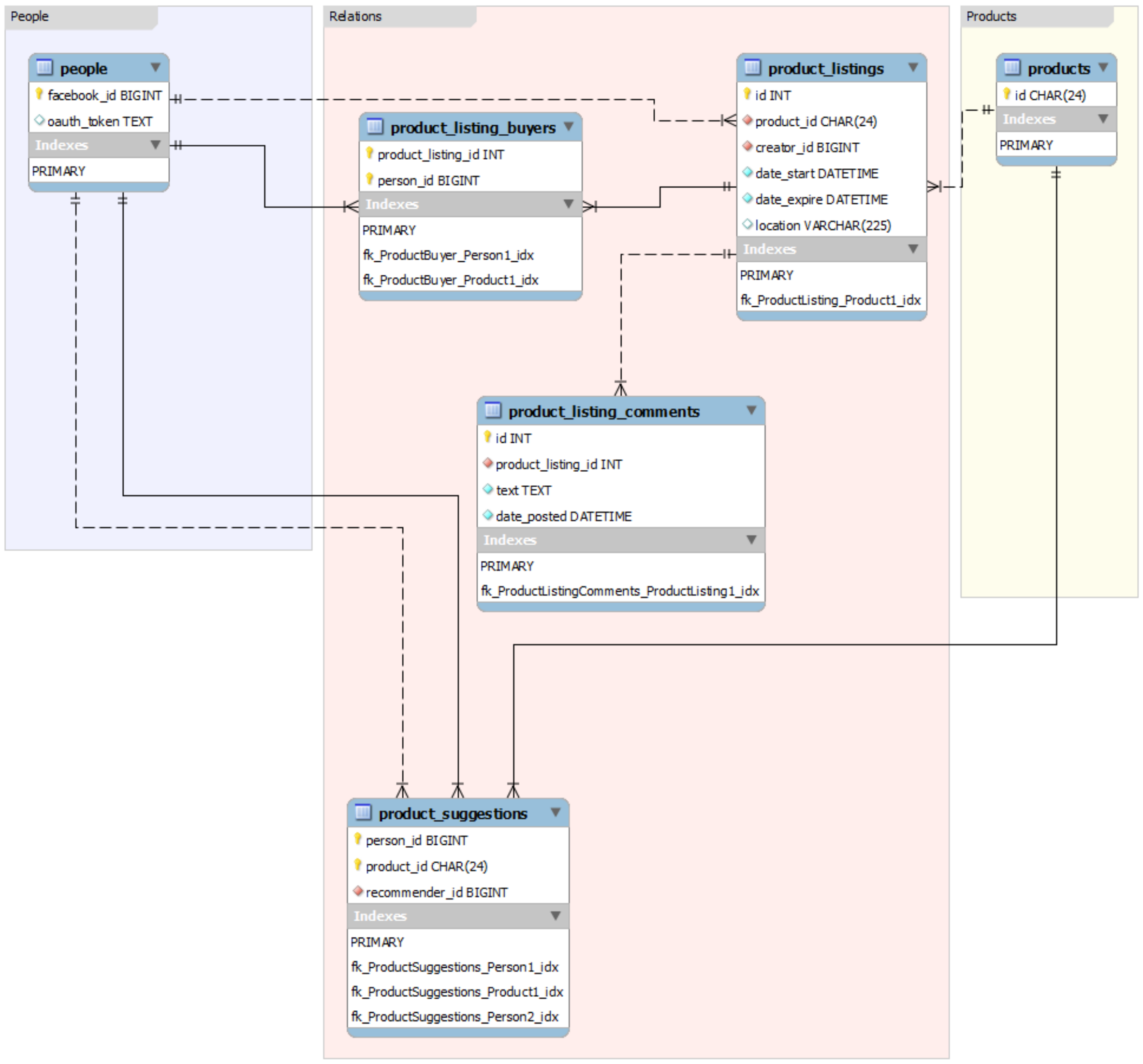
## *Aspirations 3.5(Bonus): What are the pros and cons of each method of visibility control? When should one use the Javascript method and when should one use the PHP method?*

Depending on the sensitivity of the information that's being displayed, there might be a difference in the approaches.
Because the JavaScript approach will require the data to actually be sent to the client, if it is sensitive (e.g. a password, should you be sending it to the client, for reasons I can't imagine why) the client would be able to see it by viewing the page source even if he does not have actual access to the data.
Using the PHP method, if the user does not have access to the data, he would never get to see it, short of directly trying to gain access on the server.

## *Aspiration 4.5(Bonus): What is the primary key of the home faculties table?*

Both the `matric_no` and `faculty` columns form the primary key. This is a *composite key*. There is no way to isolate any row unless both the matriculation number AND the faculty ID is specified. This can be useful for storing the entrance into the programme, e.g. when you take a minor in another degree where you might "matriculate" after your first semester. There is no way to determine when you "matriculated" unless you specify BOTH the matriculation number and the faculty you matriculated to.

## *Aspiration 5 - SQL Schema*



## *Aspiration 6 - Tell us some user queries(at least 3) in your app that needs database access.*

## *Provide the actual SQL queries you use and explain how it works*

The database schema we have is relatively simple, and SQL is abstracted away by our backend framework, CakePHP, to higher-level constructs. We will attempt to reconstruct the queries here, but they might not be the exact ones used by the framework.

### Listings which I created.

```
SELECT `ProductListing`.`id`, `ProductListing`.`product_id`, `ProductListing`.`creator_id`,
`ProductListing`.`date_start`, `ProductListing`.`date_expire`, `ProductListing`.`location`,
`ProductListing`.`friends_only` FROM `cs3216_fbapp`.`product_listings` AS `ProductListing`
WHERE `ProductListing`.`creator_id` = <my fbid>
```

Translation: Find all listings which I made. Basically a simple selection, based on Facebook II

### Listings which I can see.

```
SELECT `ProductListing`.`id`, `ProductListing`.`product_id`, `ProductListing`.`creator_id`,
`ProductListing`.`date_start`, `ProductListing`.`date_expire`, `ProductListing`.`location`,
`ProductListing`.`friends_only` FROM `cs3216_fbapp`.`product_listings` AS `ProductListing`
WHERE `ProductListing`.`product_id` IN (<product matches>) AND
((`ProductListing`.`friends_only` = 0) OR (`ProductListing`.`creator_id` IN (<long list of
FBIDs>)))
```

This is not trivial because “friends” is a dynamic concept. People can mark listings as friends-only. Furthermore, the friends list is not on our server, it’s on Facebook’s!

### Who wants to buy what.

```
SELECT `Buyer`.`facebook_id`, `Buyer`.`oauth_token`,
`ProductListingBuyer`.`product_listing_id`, `ProductListingBuyer`.`person_id` FROM
`cs3216_fbapp`.`people` AS `Buyer` JOIN `cs3216_fbapp`.`product_listing_buyers` AS
`ProductListingBuyer` ON (`ProductListingBuyer`.`product_listing_id` = <id> AND
`ProductListingBuyer`.`person_id` = `Buyer`.`facebook_id`)
```

This is a many-many relationship between the buyers and a listing. This keeps track of who’s joined in on a particular listing, and who hasn’t.

## *Aspiration 7 - Show us some of your most interesting Graph or FQL queries. Explain what they are used for (2-3 examples)*

/<fbid>?fields=currency To see what the user set his home currency to, to display prices in his preferred currency.

me/crowdbuyfb:want_to_purchase [POSTDATA: item=http://fb.sapuan.org/listings/<listingId>] To post the “Want to purchase” action to the user’s Graph as an Action. This is automatically aggregated as a Collection

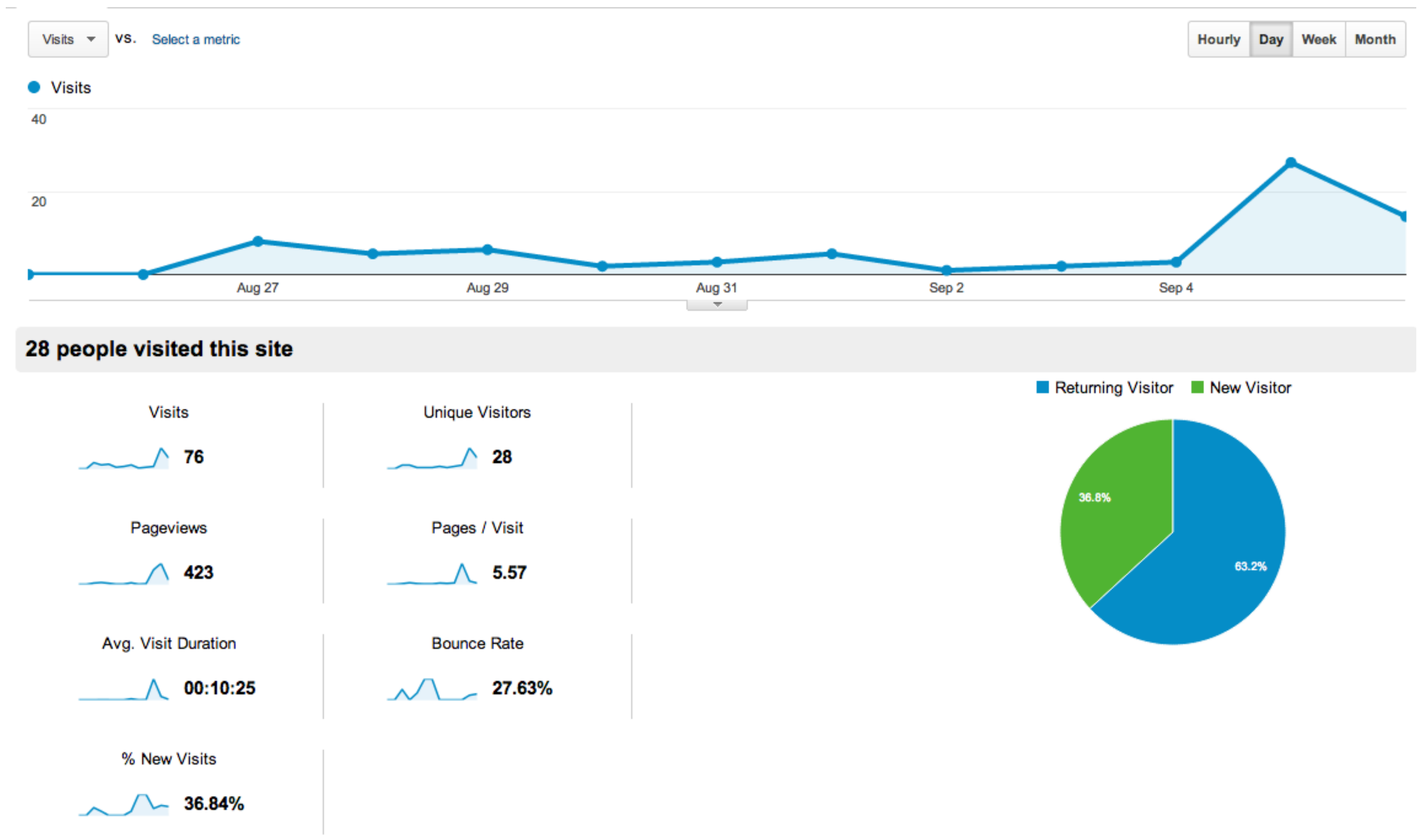
## *Aspiration 9 - Convince us why you think the present location of the <like> button is the best place you should place the button*

Currently, CrowdBuy’s Like button is placed at the top right below the logo. We believe that by placing <like> at the top, no one would miss sight of it and would be aware of it’s existence. Since the success of CrowdBuy is largely dependent on it’s number of your friends that use it, we want to have social sharing buttons that are easily accessed.

*Aspiration 11 - Explain how you handle a user's data when they remove CrowdBuy. Do also include an explanation on whether you violate Facebook's terms and condition.*

When a user removes the app, we will remove all traces of their fb_id and OAuth tokens. Facebook Data Use Policy dictates "... Delete all the data you receive from us… " and we believe that we are in compliance of their policy.

*Aspiration 12 - Embed Google Analytics on all of your pages*



**28 people visited this site**

| | |
|---|---|
| Visits | 76 |
| Unique Visitors | 28 |
| Pageviews | 423 |
| Pages / Visit | 5.57 |
| Avg. Visit Duration | 00:10:25 |
| Bounce Rate | 27.63% |
| % New Visits | 36.84% |

*Aspiration 13 - Describe 2-3 user interactions in your application and show us that you have thought through those interactions. It would be great if you could also describe other alternatives that you decided to discard, if any.*

**1. Joining in on an existing listing, posted by someone else.** The user sees the listing on the main page, and clicks the green "I'm in." button.

This interaction is incredibly simple, and is intended to be. The entire app's goal is to make it easy for users to get their friends to join in on a group purchase, and making that action only require a single click (from the app's main page itself!) was the best way to do so - it removes any friction or barriers to entry, as it is incredibly easy both to do and undo, in the same way the "Like" option on Facebook posts is.

An alternative way of doing this is from the app's individual listing view, which is seen when the user clicks on the listing (or when they are brought to it from a notification.) This alternative is still in the app now, but is not the main way of carrying out this interaction. We could have made it so (and not had the "I'm in." button appear on listings on the main page), but this would have meant users would have to go through an extra step just to carry out the core task they were expected to do most frequently.

**2. Adding a new listing.** The user searches for the product they wish to create a listing for. Noticing it in the results that appear, they click on it and are presented with a form prompting them to specify the location to which they will be having it shipped, and the date by which they will stop accepting other buyers. Their location is, if specified on their Facebook profile, already filled into the shipping location field.

Just as above, we wanted to make this process very easy and quick for the user. The number of clicks required to create a new listing once the desired product has been found is reduced to a very low number: one to bring up the form, a couple to choose the expiration date (using a convenient date-picker), and another to submit. By using the user's Facebook location to automatically populate the shipping destination, we can even remove the need for the user to type any information - the only input needed from them is the expiration date.

This process could have very easily been done by introducing another view that would have displayed more detailed product information, and making the user bring up the new listing form from there. However, this would only lengthen the process unnecessarily: CrowdBuy is not a shopping app, and so we don't need to focus too much on product information - our users already know they want to buy something, and are just looking for others who are interested in that same product!

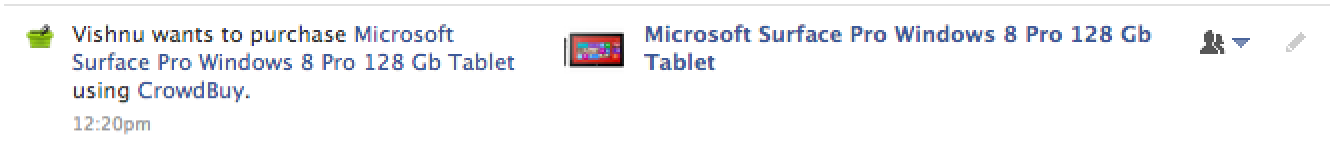
### *Aspiration 14 - Show us an interesting DOM manipulation through jQuery that occurs in your application.*

Our application uses the Backbone.js library to handle the various views, which itself delegates to jQuery. As a result, even though we don't use jQuery directly all our view manipulation is run through it.

This has allowed us to make our app run entirely on a single-page - the main listing page, search results, and item listings are all generated through JS from HTML templates, and swapped in and out of the DOM. This means users don't have to sit through page loads each time they move to a different part of the application, giving a seamless look and feel.

### *Aspiration 15 - Describe at least one Action and Object that you have created for your application and why you think it will create an engaging experience for the users of your app*

Open Graph Action: *Want to purchase*
Open Graph Object: *Item*
Whenever a user adds or pledges for a listing, CrowdBuy posts(pending permission) a OpenGraph(OG) story on their feed of the form:


Vishnu wants to purchase Microsoft Surface Pro Windows 8 Pro 128 Gb Tablet using CrowdBuy.
12:20pm
Microsoft Surface Pro Windows 8 Pro 128 Gb Tablet

When Vishnu's friend sees that Vishnu is interested in buying a Surface in a few days, Joel might be interested too and he can access CrowdBuy through the OG story and commit to buy a Surface too. OG stories such as these make perfect sense for CrowdBuy as our app is all about getting your friends interested in the products that you want to buy so that everyone can save money.

### *Aspiration 16 - Describe how you have integrated with Timeline using Open Graph Collections*

Open Graph Collection: *Recently Wanted Gallery*

CrowdBuy’s Open Graph collection allows users to show products they have recently “wanted”, in a nice gallery view.

## *Aspiration 17 (Bonus) - Describe 2-3 uses of animations in your application. Explain what kind of value do they add to the context which they are applied. Highlight your more innovative animations. Aspiration 19 (Bonus) - Tell us how you have made use of any existing JQuery plugins to enhance your application functionality*

Our app makes use of Bootstrap’s modals, which are a jQuery plugin. These modals are used to allow users to input and view information without needing to entirely leave the context they were in (for instance, adding a listing from the search page is done through a modal). Our privacy policy is also displayed through a modal.

The modals appear and disappear using a subtle animation in which they fade in/out while simultaneously sliding in/out from the top of the screen. At the same time, the page has a transparent overlay applied to it to temporarily de-emphasize it as background content.

These modals and animations make the application feel more dynamic, as opposed to being a simple set of static pages which come one after another. We can also save on having to reload (and in the process regenerate) views each time a user completes an action through their use, as we don’t have to switch to a different page for these tasks.

Another use of animation in the app can be seen when triggering a search for an item from the main page - the search results page fades in to replace the main page. This subtle animation helps prevent the appearance of the search results from being too jarring.

## *Aspiration 18 (Bonus) - Describe any cool utilization of AJAX in your application*

As we aimed to make our app run from a single page, we use AJAX quite extensively throughout the application. For instance, submitting a search query and retrieving its results are done through AJAX, allowing the app to respond quickly to user input. This is especially useful as we rely on a third-party service (Semantics3) for product listings, and their API takes a number of seconds to respond; if we had instead done search synchronously (i.e. rendering the page on the server and returning it to the user), this would cause a very noticeable delay between the user submitting their request and being able to view the results. Thanks to AJAX, we can instead switch to the search results page immediately and display a loading bar, showing the user that the application has indeed responded to their input.

## *CrowdBuy's Pitch*

Buying items from overseas is a huge pain, as shipping costs can often come close to or equal the actual cost of the product itself. Buying in bulk makes a lot of sense under these circumstances, but coordinating such efforts isn't an easy task - it's difficult to find people who want the same thing, and even harder to get simple information out to all of them at once: what you're buying, how much it is, and so on.

With CrowdBuy, it's incredibly easy to solve these problems.

Users can find almost any product in the world using CrowdBuy Search, thanks to our integration with Semantics3. From there it's a simple process to get your friends involved: post a listing, and they'll see it appear in their Facebook activity feed. And since we're using Facebook to begin with, you can even share the listing directly on your feed if you want to give them a little nudge.

As a potential buyer, it's even easier to get the information you need to make your decision: are you in, or out? All item details are available right from the main page (without any additional clicks): what the product actually is (with a convenient picture), when it'll be bought, where it'll be shipped to, and even the price (automagically converted into your own local currency.)

CrowdBuy. It's group buying made easy.