OpenKart 2.0 : Collaborative Shopping

Akshay Jain
NC State University
Master of Computer Science
Raleigh, North Carolina
Email: anjain2@ncsu.edu

Jithin John
NC State University
Master of Computer Science
Raleigh, North Carolina
Email: jjohn3@ncsu.edu

Surabhi Sudame
NC State University
Master of Computer Science
Raleigh, North Carolina
Email: ssudame@ncsu.edu

Zaheen Khan
NC State University
Master of Computer Science
Raleigh, North Carolina
Email: mkhan7@ncsu.edu

***Abstract*—With the advent of technology, online shopping is considered more of an essential than a luxury. However a major hurdle that most consumers face today is that shipping or delivery charges for a small one time order is prohibitively high and coordinating and planning grocery shopping to meet the minimum order limits of most online shopping platforms is a hurdle. To solve this problem people tend to collaborate with their friends or buy unnecessary things just to exceed the free shipping amount. At times, people altogether cancel the plans to buy some nonessential goods or make the trip to the brick and mortar store to avoid these extra charges. We are looking at an innovative approach to solve the above challenge by reducing the cumbersome task of finding people in nearby area and providing a single platform to resolve the above problem.**

***Keywords*—** Collaborative shopping, Android app, Notifications, Firebase,GeoFire, GeoLocation, NoSQL, SMS, authentication, Realtime chat

I. Introduction

Online shopping has become an integral part of the lives in the twenty first century. Online content hosting platforms and traditional shipping companies have formed a network that moves billions of items across the face of the planet on a daily basis. The shipping companies uses advanced algorithms that can best optimize delivery routes and timings to economically move packages. However this becomes difficult in a several cases like grocery shopping and rural areas with lower frequency of deliveries. The shipping optimization systems face the biggest challenges in predicting shipping patterns or coordinating shopping schedules of people. This results in higher cost in shipping part of which is levied as a fees on the consumer. One idea this solution attempts to take advantage of, in order to solve this problem is the power of crowd sourcing. The idea is to provide a platform to empower the consumers to anticipate, communicate and coordinate the shopping activities. This allows a crowd controlled platform that can coordinate the shopping activities to best utilize the shipping facilities to best place the order. This allows the consumers to make monetary gains by coordinating the shopping activities to incur least shipping charges. These solutions also helps in solving problems in commuting to grocery stores without cars or while buying things online and not be able to reach upto free shipping amount.

This takes a lot of trouble and sometimes it happens that people dont check the group or arent members of group and thus bear some loss by ordering something extra or paying the shipping amount. This solution is not well coordinated and also it takes a lot of time to gather total number of people necessary to reach the free shipping amount.

II. Literature Review

It is a well-known fact that people do not like to pay extra charges beyond what their items cost. Online shopping is a convenient way of getting our work done without the hassles of commuting and paying for the commute. If the shipping charges are beyond expectations, the overall sales enhancement due to internet may be limited. Studies have shown[1] that shipping and handling fees are the number one factor driving shopping cart abandonment.

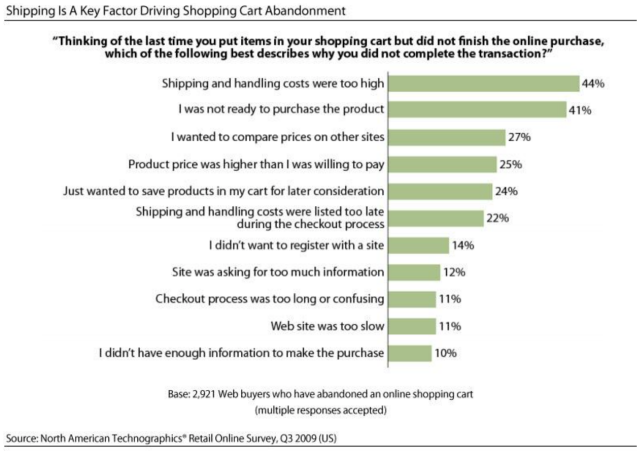

Fig. 1: Study on different issues

As we can see in figure 1, shipping and handling costs were a major factor in people abandoning their cart. This causes inconvenience to the user and loss of market to the shopping entity.

III. User Study

We had sent out a survey form to assess the extensions that we want to make to the already built OpenKart 2.0 system. Some of the questions and the analysis we have come up with is as follows:

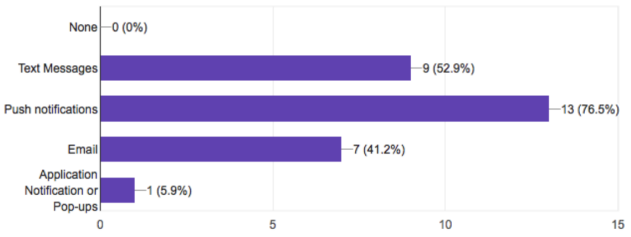

Fig. 2: User Review on Alert Methods

A. Which notification or alert method would you prefer for this app?

We found that the current application's scope did not include sending any kind of notifications to the user, which we thought that could add an edge to the user experience and make the app much easier to use. Thus we asked users this question to find out which kind of notification method would be most preferred by the users. The results as represented in figure 2, are really straight forward that the best way to notify users would be push notifications. This helps us to apply our resources in the one of these notification methods as each of these require a significant time in integrating and testing each of these notification modes, which prevents us from including all the modes of notifications in this single effort. We have decided to concentrate our efforts in designing and enhancing the application to include push notifications.

B. Would you prefer to get notifications or alerts when there is a change in the contents of the cart?

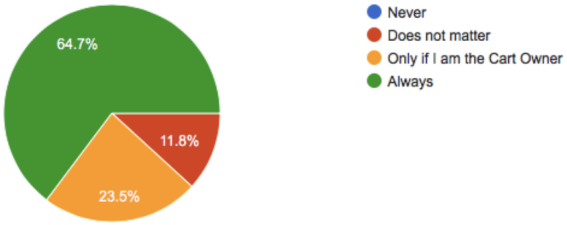

Fig. 3: Notification Preference of Users

Since we have recognized that adding notifications to OpenKart 2.0 is important, we asked this question to users to figure out when the users would want to receive a notification as we all know that extra or unnecessary notifications is something that absolutely no one wants. Figure 3 helps conclude that About 25% of the users would want notifications only when they are the cart owners and about 65% would like to receive them all the time irrespective of them being the cart owners. Even we feel that the users need to be updated all the time since their orders are also involved. This helps in tailoring the default settings of the notification feature, which help avoid negative user experience leading form unnecessary pestering notification.

C. How comfortable are you about sharing your online order details with other users?

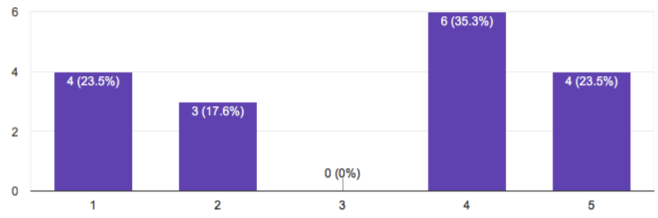

Fig. 4: User Reviews on Privacy of Cart

This question was motivated from a privacy concern point of view. We have asked this question to decide whether the items of a users cart should be made private or not. As Figure 4 shows, We have received really mixed opinions from users about how they feel about sharing their order details with fellow users of this application (who may even be complete strangers). As we have received really mixed opinions, we have decided to not move forward with this as one of our feature enhancements.

D. Would you prefer having a payment method integrated with the application?

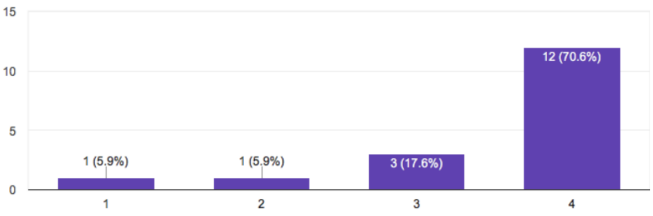

Fig. 5: User Reviews on Adding a Payment Method with App

We asked users if they would like to have a payment method integrated with this cart sharing application, and more than 90% of the users feel that is a great idea and would really love to such a feature with OpenKart 2.0 . Figure 5 shows that a lot of Users would like to have this app integrated with a payment method. Although even we feel that this is an interesting and helpful feature for this app, we have decided to not go ahead with this feature extension as we think that to implement this feature, our security has to be increased multifold as it involves a lot of sensitive data like the users bank or credit/debit card details. We feel that this does not fit the current scenario and timeframe.

E. Would you like to have the app place your order directly?

As figure 6 shows, around 40% of users are not comfortable or indifferent towards the the app auto-ordering on their behalf and also as it is not possible given our current aim and resources. We have decided to make the ordering process simple by helping and easing their process of cart creation rather than placing orders on the users behalf.

Fig. 6: User Reviews on Automatic ordering

F. What are the important feature according to you that should be added to this application ?

We have given users a set of features and asked them to rank them into top 3 according to what they think. The list of features given to the users were:

- Notify cart owner when target amount is reached.
- Integration with a payment method to share cost among collaborators.
- Suggest the second cart creator to automatically combine the contents into another cart that is already available. (Automatic Cart Merging)
- Get alerts when a new cart in available in the specified radius.
- Integrate with expense tracking apps like Splitwise.
- Integrate with online shopping websites for automatic orders.
- Allowing orders to be marked Urgent, and optimize suggestions based on Urgency and preferred collaborators

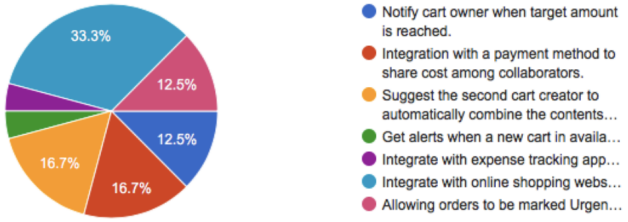

Fig. 7: Most Important Feature According to Users

As figure 7 highlights, the first choice of most of the users was to integrate it with online shopping websites so that the process of transferring or replicating the cart from the app to the website will be tedious process. The user will have to keep switching applications and then searching and selecting the items in the cart from the app on the online shopping website. This makes the user indulge in useless extra efforts.

As figure 8 shows, the second most important features according to our surveyors are providing the users with notifications and integrating a payment feature into the application. Even according to us, notifications to users should be enabled as this will increase the user experience greatly and as this app includes users buying things and eventually paying for it, we feel that the user should be continuously updated.

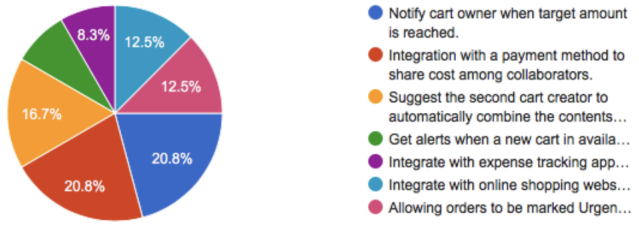

Fig. 8: Second Most Important Feature According to Users

Although we acknowledge and understand that payment integration is an important feature with regard to the purpose to this application, we have already discussed why integrating a payment feature is not feasible for us right now given the scope and time frame of this project.

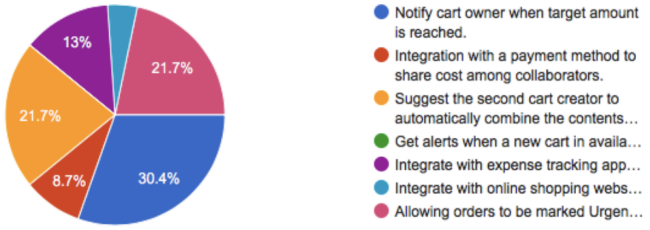

Fig. 9: Third Most Important Feature According to Users

As highlighted by figure 9, Users see that making orders of each user private and the automatic cart merging as the other important features that we should be adding to OpenKart 2.0 .

We have decided to exclude making orders completely private as this will create complications when replicating the cart from the app to the website as if the order items are kept private then how will the cart owner know what items to add into the actual cart on the shopping website. Thus we have decided to implement a part of this as we know that users value this feature. The orders will be made private to everyone else other than the cart owner as this is the only possible solution to this issue as of now.

We also want to implement the auto-merging of carts as a feature as we think that users fail to recognize that deleting their own cart and re-adding the items into someone elses cart is a very tedious and time consuming method and will eventually discourage users from merging their carts and will slow down the target completion and users will simply have to compete between one another.

IV. Existing Solution

Thus, taking an inspiration from this, we are providing a single platform in the form of an android application that would solve all the above problems and help the users to collaborate easily to solve these issues. Here the users will

be allowed to announce their purchase in form of creating a prospect order on our application. When someone is finding users to collaborate with, they will be able to set a radius and the App will display the ongoing prospect orders within their mentioned radius with amount due and approximate order date. Also additionally when users collaborate and want to communicate with other collaborators in the same order they will be able to use the inbuilt chat interface that we provide to resolve any questions they have regarding the order. Thus, we are hoping that our application would provide an efficient platform to solve this day to day problem of the users.

The first version of the application has been developed by group B as part of the Phase 1 project of Software Engineering course[4]. This mobile application is well developed with modular code base along with the extreme necessary features. A user can create a cart setting up the target, add his or her own items, and wait for other users to add their items and reach the specified target. Once the cart reaches its specific limit, the users can chat amongst themselves to decide on the time and place of delivery and collaborate on picking up their items from the place of delivery. Before creating a new cart, a user can check for existing carts nearby using a convenient slider for radius adjustments. If found a cart that the user is comfortable with, the user can add his or her items to the cart along with their price, the link of the food store for the item, etc. The price of his or her items is then added to the cart and once the total price reaches the target amount, the users can collaborate on the chat window and place the order to get their items respectively.

V. Enhancement Plan

We had an open-ended discussion on how the application can be enhanced to make it more user-friendly and rich in features. The following were some of the items discussed:

1) Notification to cart owner when target amount is reached: The cart owner currently has to keep track of the total amount in the cart and whether it has reached the specified target. This is cumbersome and requires unnecessary manual effort. The cart owner may either check it multiple times with no change, or forget about it and not check it at all. This will make the collaborators wait on him forever. Both of these are bad user-experiences. Thus, we concluded that a notification should be popped up on each addition / change to the cart so that it is easy for the owner to track the cart. At what instances and who the notifications should be sent to was left to the general preference of users and was included in the survey as a question to determine the same.

A. Privacy concerns

Currently, the items in the cart specific to individuals are openly viewable by all collaborators in the cart. This raises privacy concerns, as people may not be extremely comfortable with openly showing what items they are purchasing. Of course, all the items will have to be visible to at least one of the cart collaborators, because he or she will need to order them. However, it is not necessary for all the others to view each others items. They only need to see the total cost and the target amount of the cart. However, before deciding on implementing the same, we wanted to validate how comfortable or uncomfortable people will actually be in sharing their items publicly. This was thus included in the survey questions.

B. User alerts for specific stores

There are a few stores that users visit multiple times in a week for groceries and items that are used every day. For such stores, users should be able to set alerts that notifies them of any carts created. The user can take advantage of these notifications and avoid multiple trips to these stores by being informed about the new cart created.

C. Seamless merging of carts

When the user base grows and the number of carts increase beyond measure, it may be cumbersome for users to check the available carts and decide on the best cart to add their items to. This might become inconvenient for the users to think and choose from a lot of options. Instead, if the user can simple create his or her own cart with his or her items, and the algorithm finds the optimal carts to add the items to and notifies the user of the same. The user will then have the choice to merge his or her cart to the suggested cart on the click of a single button. This eliminates the need for users to manually look for carts and decide from them. Also, a periodic optimization can suggest carts to different users if better carts become available even after they have merged to another cart, if both the carts can reach their target amount with such an optimization.

D. Optimized cart suggestion

We will be enhancing current cart searching algorithm. Currently the app shows all the open carts based on radius. We will be optimizing the algorithm to find the best matching carts based on order priorities and user preferences. New application will allow user to mark the priority for an order and select with whom they wish to share the cart for example share with only contacts, share with previous collaborators or share with anyone. Based on these preferences the app will show the best available carts. For example, if an order is marked urgent, then list of available carts can also include new collaborators.

As integration with the store is currently not possible due to lack of open APIs, we can periodically crawl through their website to sync the product details with our servers. This can work because a slight change in the details at times may not hamper the usability of our application.

E. Persistent groups

Once users get comfortable with ordering with a particular group of specified people, they should be able to order with them again repeatedly with minimal effort. Thus, users should be able to save particular groups for ordering without looking for such carts.

F. Auto-fill product details

Currently, users adding to an existing cart need to enter a lot of details for each product to ensure that the person delivering the goods orders the correct product. Some products are available in different sizes, from different companies, with different prices. Thus, all of these details have to be entered manually and correctly from the actual store website to the existing application. If there is an integration between our application and the store website, the users should be able to select the correct product on the application itself. This will help them to add the appropriate item to the cart with ease.

VI. Implementation Details

A. Inventory extraction from popular stores

As mentioned above, since popular stores do not have open APIs that expose their data, our best bet right now is to scrape data from their websites. We will save this data on our database servers and auto-fill the product details for the users.

There are a number of popular crawling and scraping frameworks available in a number of languages. Some of the most powerful crawlers are available in Python and PHP. Scrapy[2] is one such framework. It is supported by an active community with which we can build our own scraping tool. In addition to scraping and parsing tools, it can easily export the data it collects in a number of formats like JSON or CSV and store the data on a backend of your choosing. It also has a number of built-in extensions for tasks like cookie handling, user-agent spoofing, restricting crawl depth, and others, as well as an API for easily building your own additions.

B. Private cart

Each cart is mapped to an owner, and each product in the cart is mapped to a user. The product details in the cart for all users should only be visible to the owner of the cart. For this, while displaying the cart contents with details, the content should be visible only if one of the following two checks passes:

- The product of the cart belongs to the user
- The user is the owner of the cart

This is currently based on the assumption that the owner of the cart will be the user delivering the goods. If not, another field should be introduced associated to each cart: deliverId. The second check will then change accordingly.

C. Notification and Alerts

There are a number of solutions available for sending notifications to Android phones. We decided to use Firebase Cloud Messaging (FCM)[3] realtime database, hosting, authentication, etc.

FCM is a cross-platform messaging solution that lets you reliably deliver messages at no cost.

Using FCM, we can notify a client application that new email or other data is available to sync. We can send notification messages to drive user re-engagement and retention. For use cases such as instant messaging, a message can transfer a payload of up to 4KB to a client app. It's key capabilities are:

- Send notification messages or data messages - Send notification messages that are displayed to your user. Or send data messages and determine completely what happens in your application code.
- Versatile message targeting - Distribute messages to your client app in any of 3 waysto single devices, to groups of devices, or to devices subscribed to topics.
- Send messages from client apps - Send acknowledgments, chats, and other messages from devices back to your server over FCMs reliable and battery-efficient connection channel.

VII. Conclusion

OpenKart 2.0 is oriented towards enhancing ways in which users can better connect with the right group users to collaborate with to coordinate their shopping activities. The features like inventory extraction is geared towards making it easier for users add new items to existing carts and also for the cart owners to easily place orders using links to the right products. This newer version of the app is also designed to have better privacy settings like hiding the cart items from users other than the cart owner. Use interface is also another important area that will be enhanced using features like push notification. Overall the newer version of this app is designed with the user experience at the highest priority.

References

[1] *Smarter Strategies For Free Shipping - Understanding The True Costs And Benefits to Retailers*, https://www.ups.com/media/en/Smarter_Strategies_for_Free_Shipping.pdf

[2] *Scrapy - An open source and collaborative framework for extracting the data you need from websites*, https://scrapy.org/

[3] *Firebase Cloud Messaging*, https://firebase.google.com/docs/cloud-messaging/

[4] *OpenKart: Collaborative Shopping*, https://github.com/srujanb/OpenKart/blob/master/MarchFinalReport-OpenKart.pdf