IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 5, Ver. II (Sept - Oct. 2016), PP 54-57
www.iosrjournals.org

Personalized Offers in Local Shops Using Client-Server"

Aritra kumar, Kuntal Biswas

Abstract: *Everybody loves shopping, in one way or the other. Some prefer clothes, while others may spend hours, searching for that perfect gadget. Shopping uplifts mood, nobody is unhappy after they have bought something that they like. Not so quite recent boom in online shopping portals is the result of the same mindset, i.e. shopping makes people happy, so the experience should be made effortless, secure and accessible to almost everybody with an internet connection. It isn't uncommon to crack a more than decent deal online, using their coupons and discount structure. As a result, the brick-n-mortar stores are facing a very stiff competition, in attracting customers, in developing countries like India, where the public is very cost conscious.This recent boom in online shopping portals has resulted in a sizeable number of small shops going out of business. They cannot compete with online mega stores with their heavy discounts and frequent deals. It becomes difficult to keep track and manually manage deals, every now and then. They do not have a huge sale amount to offer uniform deals on every item available, nor do they have the man power to track every detail of units sold etc.*
Our project aims to deal with this issue, and subsequently provide the small businessmen, a tool to manage their merchandise, and the shoppers to have a seamless experience while shopping. The following sections describe the structure of our project, and provide explanations and results wherever applicable. The introduction section lists the reasons why we felt that this project was necessary. It also contains the outline of the structure of the program. The methodology section describes the working structure, and the communication methods. The result section provides a window to the results and screenshots of a prototype model.

I. Introduction

Before the advent of online shopping giants, the market was a busy and lively place. A constant stream of customers, a loyal customer base, and good business throughout the year. Life was good. That is, until people found the ease of shopping at home. No browsing through aisles to find the needed product, good discounts, return policies etc. spoiled people to the extent where going out to shop became hectic. Some vendors even provide home trials for clothing and accessories. It isn't too difficult to imagine the disappointment of finding out that the desired product is unavailable after browsing through a local shop for hours.

What makes online shopping so convenient? Is it their product search feature, or their discount structure, or their overall customer relation? It is a bit of all. Why not give the same advantage to local shops as well? After all, a little competition is good for both the businessmen as well as the customers. In no circumstance do we aim to challenge the established giants, but through this project we do try to address their shortcomings, where the local businessmen get an opportunity to step in. The biggest gripe while shopping online is their delivery time. Buying stuff is easy, waiting a week for it to arrive isn't. Whereas, local shops provide immediate access to their merchandise. We are hoping to find a sweet spot in between the two.

To design this project, we have used a number of tools. The language used is python, since I feel that this is one of the easiest language to code with. With a syntax that necessitates indentation, the codes are easy to distinguish, and are as close to formal algorithm written in English language, as possible. A module named sqlite3 has also been used, to maintain separate tables for articles under different categories. Different tables have been maintained for distinct categories, to make it easier to search for items, and maintain tags, that are later used to calculate discounts. The client server architecture has been implemented using Bluetooth. A module named PyBluez has been used to design the client server architecture. The project, in its basic form, is designed to run on two separate machines, although it can be expanded to a number of devices if applied on that scale. The user data is stored in files, and lists are used to temporarily store some of the data. The user interface is designed using Tkinter module. An example graph has been plotted and displayed using matplotlib module.

II. Methodology

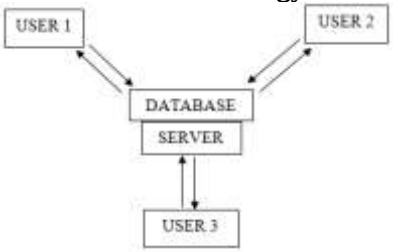


Fig. How the communication works with the central server

This project is designed to work in a client server architecture. The shop node serves as the general, centralized server, while the clients are the shoppers, querying and asking for services. The main features of this design are as follows:

1. Cart: A facility for maintaining a shopping cart has been provided, where customers can save their choices, even before going to the shop. They are free to manage it completely offline. The cart facility is a standalone feature in the customer devices. It allows the users to save a list of items that they would like to buy. It is never exchanged between devices, and is safe from snooping from outside the device. It is inspired by real life grocery lists, only valuable to the owner, and completely useless to anyone else.

2. Search product: While in the shop, the customers can search for their desired products, which returns the product and prices. The search section effectively uses the client server system. The customers can query the server, if an item is available in their store or not, for which, the server replies with a suitable reply. An 'add to cart button' has also been provided, after a successful search, where the customers can add the item to their carts.

3. List management: The shopping list saved by the customers can be viewed on their devices. Since this data is stored on their respective devices, this information is completely private. No snooping can be done on these, since these are never connected to the server directly. The list management section manages the items that have been added to the cart by the user. The idea is to save the search data to the customer device. When the user presses the 'manage list' button, this very information is searched to find the items and respective prices, and print them in a readable tabular form. The total amount is also provided for the customer in the same window.

4. Offer zone: A provision for offers and discounts has been reserved, where every user gets a personalized discount based on his/her shopping behavior. The offer zone is the strong point of this project. Rather than having a fixed discount percentage or criteria for everybody, a dynamic calculation is done to calculate discount for a person. This discount is related to the number of items bought, in a way where, the more a person shops at an outlet, the more discount is offered. One more point to note is that, doing this for every item will incur heavy losses to the seller. So, categories have been set aside, where an item must exist in one or more categories. When a person buys an item of a particular category, the category_counter is incremented by one. At the time of checkout, discount is offered on the category of items whose counter has the most value.

A simple observation would report that if the discount is directly proportional to the number of items bought, for a certain number of items the discount will be equal or more than the selling price. We do not want that. Instead, we have subtracted the base discount offered from the maximum profit earned by the seller. Then we have subtracted the reciprocal of the number of items bought from the amount, which gives us the final discount percentage.

The logic behind this is, that the discount ranges between two values, max_discount and min_discount, increasing with the number of items bought, but staying in its limit. It never increases beyond that point of max_discount. The seller is free to set the range as long as it is lower than the maximum profit earned. This whole calculation is done at the server side, hence, the customer is unaware of the behind the scenes criteria and calculations. He gets the final discount amount on his device.

The mathematical formula used is:

$$\text{Discount} = \text{max\_discount} - \frac{\text{Range of discount}}{\text{Number of items bought}}$$

Where,

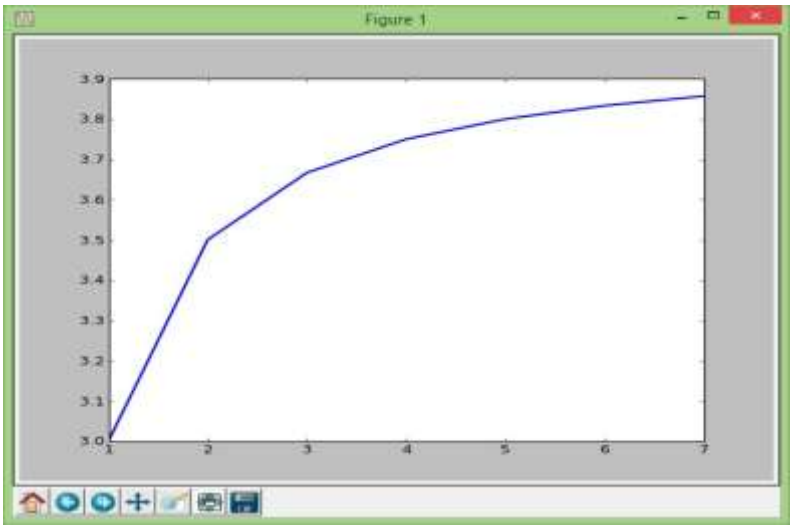
Discount = Discount is the discount percentage,

Range of discount = Max_discount – Min_discount,

Number of items = The number of items, of a particular category, bought by the user

For example, if the seller has bought an article for 100 and was selling it for 110, he was earning a maximum profit of 10%. Now, the seller offers a maximum discount of 4%. This amount resides on the server, and is not visible to the user. So, the base_discount becomes (4 – 1) % = 3%, assuming that the seller has decided to set the range of discount to 1 which is the advertised base discount percent. This will increase with the number of items bought. Now, suppose a person buys 5 items, the discount % is 4 - 1/5 % = 3.8%. Similarly, if the person buys only two items, the discount % becomes 4 - 1/2 % = 3.5%, which is lower than the discount offered when he bought 5 items.

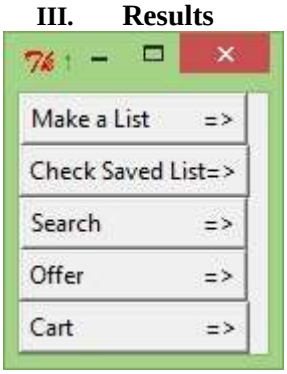
Following is a graph plotted for the discount percentage, against items bought. For a sample set of values, the items bought are a set of whole numbers ranging from 1 to 7. Items bought are plotted on the x-axis, and the discount percentage is plotted on the y-axis.



If we look at the curve, it saturates at base_discount+1, i.e. at 4%. This is the main theory behind the personalized discount.

5. Product finalization through device ID: Customers can finalize their products and check their final billing amount before going to the checkout counter. It helps manage their budget efficiently, and saves the hassle, if by chance they forget any item, since this list displays all the items that they have added to the cart themselves. The product finalization occurs at the checkout counter. The checkout counter at the shop matches the items there with the items in the list. This is done manually, for the time being. Although, at this point, the category counters are written in files, mimicking the concept of cookies. This cookie file is later referenced when the person visits next, and a discount needs to be calculated. This portion is done at the client device, allowing the server to run void of dependency files. This also makes it light weight, and communication remains fast.

III. Results

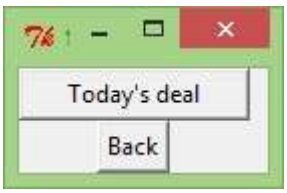


Op1. This is the main window. It provides access to the functionalities listed in the window.

Op2. This is the menu to add items to the shopping list.

Op3. This is the window to search the availability of an article.

Op4. A sample shopping cart, with the checkout button. This button_function checks the tag of items and writes them to a file, with the tag_counter.



Op5. The deal window, clicking on Today's deal shows the current deal, based on the details in the cookie file.

Op6. A sample offer for a customer.

IV. Conclusion and future work

The purpose of developing this project, was to provide a cheap way for the local shops and businessmen to provide the ease of online shopping, and the access of locally available merchandise, at the fingertips of consumers. Apart from standing out among fellow businessmen, this also relieves the employees from a lot of retail related tasks, like product enquiries and discount calculations. All they need to do is, categorize the goods and interact politely with the shoppers. Although, this method still requires people to get out of their homes, and go shopping, we still think that it is a perfectly accessible solution now. After all, a little exercise, and a little adventure never hurts anybody. We hope to have attained this purpose with this project.

Some work still needs to be done on this project. Namely, a security system needs to be implemented, to ensure customer privacy. Also, the users must be denied access of the cookie files, to ensure that no external modification is made on the counter data. Although a database could be used for this purpose, we still feel that it is an overkill for this small purpose. We have limited the server to a text based service, but a good intuitive user interface could be developed for the server as well, that might aid even novice employees to use the software efficiently. Although, the current system works fine, and fulfils its most basic needs.