

BuySmart - connecting offers and consumers

Henrik Holm and Ulf Aslak

Personal Data Interaction - DTU 02808

Abstract. This study investigates the viability of a mobile application that bridges the gap between consumer and offer, by letting the user know where items on a shopping list can be purchased on sale. The app is designed to record the user's offer purchasing patterns and present the data to the user in order to increase awareness of own habits. The study finds the application technically feasible and, through evaluation, relevant to the needs of four qualitative test subjects and over 100 respondents to a survey.

1 Introduction

In spite of the tireless efforts of retailers to draw attention to offers and attract customers to their stores, most people are generally unaware of offers until they come across them physically - and with good reason. For many people, investing the necessary amount of time to save money in e.g. everyday grocery shopping is untenable and simply not worth the savings. This, however, does not mean that the average person is not interested in making a good deal.

This study analyses the potential for improvement in the interaction between retailers and consumers, facilitated by offers. The approach is largely exploratory and relies on cyclical prototyping and evaluation of a mobile application. At its highest level the application is designed to act as a facilitation mechanism for symbiotic interaction between stores and consumers. The app is given the name *BuySmart*. At a lower level BuySmart works as any traditional shopping list app. In the above layer, BuySmart records user location and items entered in the shopping list, and shows where the items on the list can be purchased on sale. In its top layer, BuySmart records the users offer shopping patterns and maintains visualisations of rolling month/week statistics, giving the user an overview of own purchasing patterns.

In order to verify, first the relevance of the high-level solution, and secondly the viability of the lower-level prototype, two key hypotheses must be addressed and proven false:

1. New customers **cannot** be motivated to interact with offers even if they are given the appropriate platform.
2. Feeding back personal offer purchase statistics to a user has **no** influence on purchasing patterns, and adds **no** layer of value that maintains the user's interest in the app.

This study will not attempt to conduct broad analysis in order to prove their opposites true, but instead take the first steps to falsifying them and thereby gradually justifying the existence of BuySmart. It will first consider the ways in which it stands apart from related work and explain the contextual problem and some of the analytical methods used to better understand it. The design and implementation will then be explained after which the methods of evaluation are presented. Finally the results from the analysis and the evaluation are presented and discussed. The project is concluded with comments on the hypotheses, and ideas and suggestions for furthering the work are presented.

2 Related Work

BuySmart uses the eTilbudsAvis API[1], hence it is highly related to the eTilbudsAvis app. The two apps addresses the same problem in the market, but with different approaches. In the BuySmart approach the information served by the API is combined with the information in a shopping list. It furthermore adds a layer of personal data interaction by recording usage data and giving feedback on habits. The current eTilbudsAvis app is not personalized in this way, and does not provide any information about shopping habits. It is furthermore not possible to make searches on more than one item at a time. Another related app is Minetilbud [2]. Both eTilbudsAvis and Minetilbud are offer-catalog aggregation apps. It can be argued that these apps are merely extension, in format, of traditional offer catalogs, hence only proposing added value to consumers that already enjoy the process of discovering offers. In order to attract the kind of users that don't enjoy this, BuySmart is the app that knows what the user needs and can present this in a simple and orderly way.

The personal data aspect of BuySmart relates in high degree to kvittering.dk [3], where users sign up with their credit cards, to receive receipts from selected stores electronically through the app. The type of data collected in BuySmart and kvittering.dk relates, but is ultimately different, since kvittering.dk records everything and BuySmart records only offers.

3 Analysis

The main problem is that most consumers are unaware of offers until they see them in the stores. Retailers, use offer catalogs and commercials to attract customers, but this only appeals to a certain group of people. Digital extensions of the catalog format does exist, but offer only limited added value to users who are not inherently interested in making efforts to browse catalogs. At the same time, the consumer group that doesn't use catalogs likely still care about offers, and by this logic there is a vacuum in the market, which can be filled by a platform that provides an effortless way of using offers.

In order to clarify a number of basic, but important, question relating to the relevance of this app, a quantitative survey was conducted. The survey, hosted by SurveyMonkey, was shared through the Facebook channels of the authors and received 157 responses over 48 hours, from people of both genders aged 18 - 32 (rough estimate based on likes received on posts and shares). For various reasons relating to bias (e.g. friendship, age, demography) the results of this survey should be trusted only as a rough outline of the true landscape that it aims to map. The survey asks 10 multiple choice questions in order to address and clarify the three more general questions:

1. How do you generally get to know about offers and how much do you care about them?
2. Do you use a shopping list and if so how, and how often?
3. Would you like to use an app that effortlessly bridges the gap between you and offers?

The full list of questions along with the survey results is available in Appendix D.

4 Prototype design

The first prototype was designed as a notification based app that sought to narrow the physical gap between consumer and offer by having the user enter a list of desired items, and send push notifications when the user was close to these items on sale (see Appendix A). This prototype was never implemented with code because it quickly became evident that it would send too many requests to the server running the API. Evaluation furthermore rendered the bearing notification feature inconsistent with user needs, and was generally characterized as *annoying*.

The current, and second, iteration of the prototype relies more on user interaction. It is a *shopping list app* that allows users to quickly gain an overview of where the items on the shopping list are on sale. In an assumed typical use-case the user will bring the phone when shopping, where the app will help check off items from the list as they are purchased (this is elaborated upon below). The app records location, time-stamp and item data for each item that is checked off. Over time this produces enough data to feed back visualizations to the user in an effort to promote a better understanding of the users own offer-shopping habits. This design is illustrated in Fig. 1. In another use case the user is at home and uses the app the prepare a shopping trip. Searches yield many variations of the queried items, where only some are relevant. Here the user could click the "+" symbol in order to replace the search term, with the found item, in order to narrow the search.

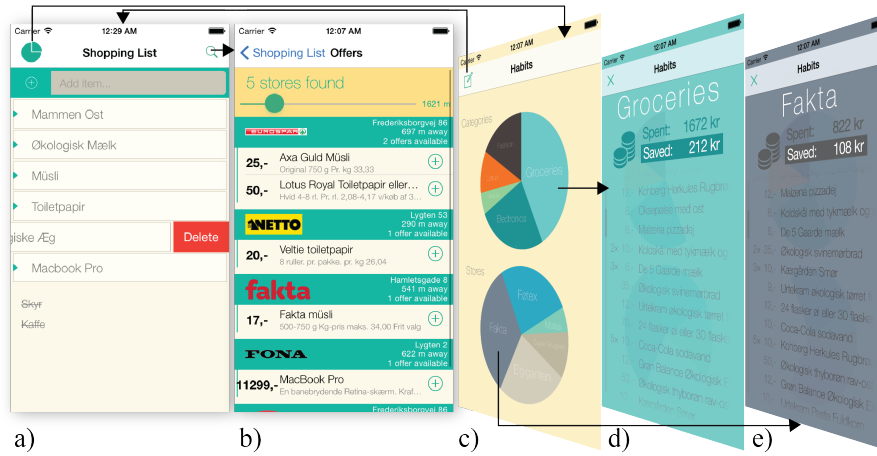


Fig. 1. Design of the second and current iteration of the app. a) Shopping list. b) List of offers on items in shopping list sorted by store. c) Habits view with pie charts showing what type of item the user buys most often and which stores the user visits most. d) Item category view showing how much money the user has spent and saved on offers in the given category. e) Store view showing amount of money spent and saved on offers in store. In d) and e) the money icon can be clicked to see an overview of total and average time spent with store/item type.

The interaction design for checking off items as they are purchased is still unclear in the prototype. Using the app now, it feels equally intuitive to check off items in the 'Shopping List' and the 'Store Offers' views. Resulting from internal evaluation, the current prevailing solution to this ambiguity problem is an option to click a store after which a new shopping list is produced,

comprising the offers in the given store as well as the items from the original shopping list. This is illustrated in Appendix C.

The design of the 'Shopping List' and 'Store Offers' views results from evaluation of early horizontally implemented prototypes. An alternative approach, which was considered early on, was to have the offers, related to each shopping list item, appear *under* the respective item, in fold out menus and thus reduce the shopping list and store view to one single view. This design is illustrated in Appendix B. The point of this design was to make it more apparent to users why certain offers were presented to them. This, however, came at the price of not being able to sort the items by store, which proved to be a common concern among potential end users.

4.1 Implementation

The underlying architecture of the design is illustrated in Fig. 2. This is implemented in iOS 8 using the Swift programming language, v. 1.2. The views in Fig. 1.a and 1.b are vertically implemented in the classes 'ShoppingListTableViewController' and 'StoreOffersTableViewController', which use the 'ETA_API' class method `getOffersFromList` to access and manipulate data from the 'ETA_SDK' [4]. The SDK, which is developed in Objective C by eTilbudsAvis, is imported into the project using Cocoapods. Because of the mismatch in programming languages, the SDK is accessed through a bridging header. The 'AppDelegate' class is generic to all iOS applications, and handles everything related to changes in the iPhone system environment, such as location and persistent storage. 'UserFeedbackViewController' is responsible for handling the views in Fig. 1.c, 1.d and 1.e, but has no implementation in the prototype. Instead this part is implemented horizontally, i.e. as images in view controllers with no underlying programming.

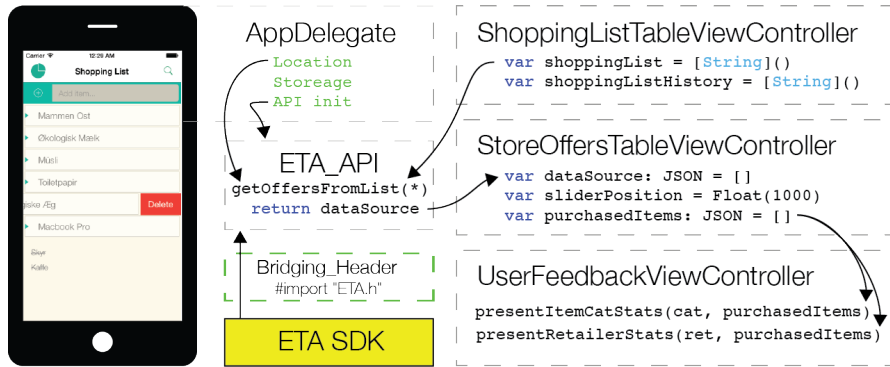


Fig. 2. The application architecture. Dashed line blocks represent classes in the program architecture. The yellow block represent the eTilbudsAvis Software Development Kit (SDK), which is accessed through a bridging header marked in green dashes. The bridging header is necessary because the SDK is implemented in the Objective C language, and the prototype is implemented in Swift. Variables and functions listed in each class are global.

The 'Shopping List' and 'Store Offers' designs, are implemented vertically because they are deemed too conceptually foreign to end users, and therefore need to be largely self-explanatory. In other words, it is necessary to have a

detailed prototype that can, in itself, communicate the concept to test subjects in order to facilitate the discovery of new unexpected use cases, design flaws and important undiscovered features, etc.. Also, it was an early priority to ensure that the 'ETA_SDK' could be used for the given purpose.

5 Evaluation

Since the app is in its early stage of development, the evaluation methods used have been mainly direct observation methods such as *Quick and Dirty* and *Thinking Aloud*. Evaluating compliance with heuristics have not been a top priority at this stage and has thus not been performed with scientific rigor, but was considered in the more informal internal evaluation throughout the design process.

The Quick and Dirty method was used initially to test the low-fidelity prototype, hence showing the first design iteration of the app to potential end-users (see Appendix A). At this point the main focus was to create a notification based app, so the purpose of evaluation was to get feedback on whether this was a feasible goal. More specific questions were also posed, such as how often the user should receive notifications. Feedback was divided, creating a demand for a more quantitative approach which was achieved through a survey.

After development of a minimum viable product high-fidelity prototype (Fig. 1.a-b), Thinking Aloud evaluations were conducted on four test persons. The evaluations were recorded using iPhone integration with QuickTime through a Mac. The Thinking Aloud evaluations had three main purposes: to test the usability of the app, to test the users ability to find value in the app, and finally to test the user's general interest in the full concept. On a higher level, the evaluations also functioned as a mechanism for discovering new potentially important/necessary features. It should be noted that these evaluations corresponded only to one of the two use cases mentioned in the Design section, namely the one where the user is at home planning a shopping trip. Further evaluation should be conducted in order to better understand problems in the design that may only become apparent during shopping.

6 Results and Discussion

Based on the results from Quick and Dirty evaluations as well as the survey which was conducted, it was chosen to pivot the app from focusing on notifications, towards integration of offers into a shopping list. The survey showed that that far more people were interested in an app similar to the second prototype than the first. Worth mentioning, is that this percentage is significantly greater than the amount of people already using the eTilbudsAvis app, indicating that hypothesis 1 is likely false.

From the Thinking Aloud evaluation a number of usability problems were discovered. As an example, the 'Store Offers' view not updating automatically upon entering or changing of the radius slider position was repeatedly mentioned as an annoyance. More general problems were also discovered, such as the need for a map showing location of stores and a menu to show more specific information about offers. It was also requested that the user could be notified on *certain* important items, should they come on sale. In general

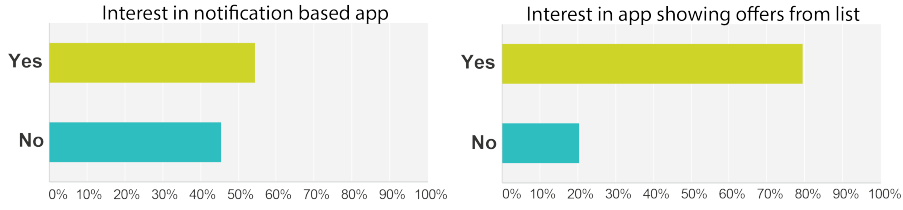


Fig. 3. Comparison of Q7 and Q9, from the survey (Appendix D).

the test persons were highly positive about the app, and seemed to understand how it could add value to their lives. The feedback integration was also reviewed positively.

Based on feedback from the survey, the results clearly show that this app could be interesting to a lot of users. 43% of the survey respondents claimed to use the current eTilbudsAvis app to find offers. Taking into account the low age mean of the test group, this high number should, however, be taken with a grain of salt. From a technical point of view, the app proves feasible.

7 Conclusions

Recalling the hypotheses constructed in the Introduction, both have to some extent been falsified. The first hypothesis stating that new users cannot be motivated to interact with offers, even if given the right platform, is disputed by the observation that more respondents are positive towards an app similar to ours, than the amount of respondents using the eTilbudsAvis app, as shown by the survey. Furthermore the second hypothesis is partially disputed by the observed excitement from test subjects, when navigating the horizontal implementation of the feedback statistics. This, however, does not prove that interest can be retained over time or that feedback may change user purchasing patterns. Proving such things would require more extensive research methods. In conclusion, the app design developed in this project has proved viable within the testing environment.

8 Future Work

A number of aspects remain unexplored in this study, both in terms of further design opportunities but also with respects to marketability. Addressing the first aspect, further iterations of the prototype must be developed in order to reflect the findings documented in this report. This also involves a more vertical implementation of the feedback design, in order to better understand how this can be used as a basis for maintaining the users' interest in the app. Another logical step would be to implement Machine Learning techniques in the program in order to suggest offers to the users based on past shopping history. Regarding the potential marketability of BuySmart, research must be conducted in order to understand the value of the data which is generated in the app, as well as the privacy issues that this creates. A realistic first approach to this would be to issue a beta version of a slightly more evolved iteration of the application to a group of test users and instruct them to use the app regularly, thereby generating a data set which would found the basis of both a market study and a privacy study.

References

1. eTilbudsAvis API, <http://docs.api.etilbudsavis.dk/v2/docs>
2. Minetilbud, <https://minetilbud.dk/>
3. Kvittering, <https://kvittering.dk/>
4. eTilbudsAvis iOS SDK, <http://engineering.etilbudsavis.dk/native-ios-eta-sdk/>

A Prototype 1

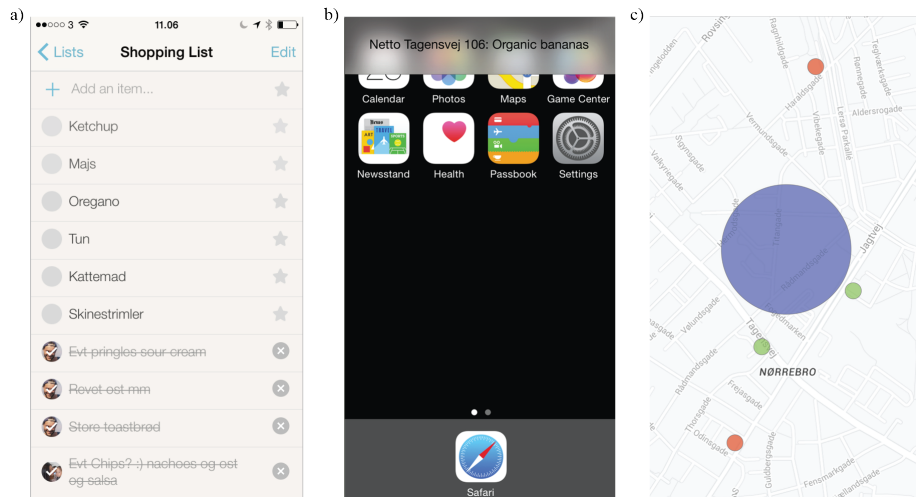


Fig. 4. a) The user types in a list if desired items. The user can mark the item as important by pressing the star-field, such that it will always trigger a notification if it is nearby. b) The user receives a notification saying that a given item is at a nearby address. c) Having pressed the notification, the user is instantly presented with a map view showing where the nearby store is, along with other nearby stores that may also have offers.

B Alternative design for prototype 2

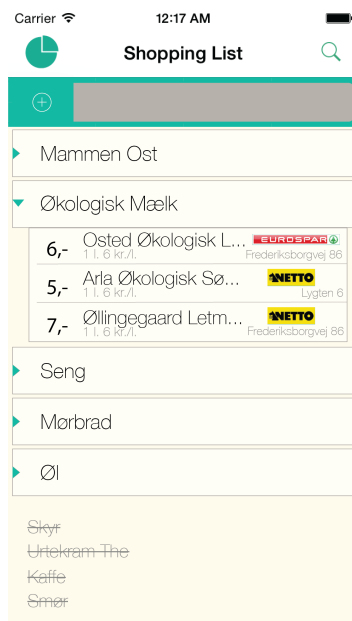
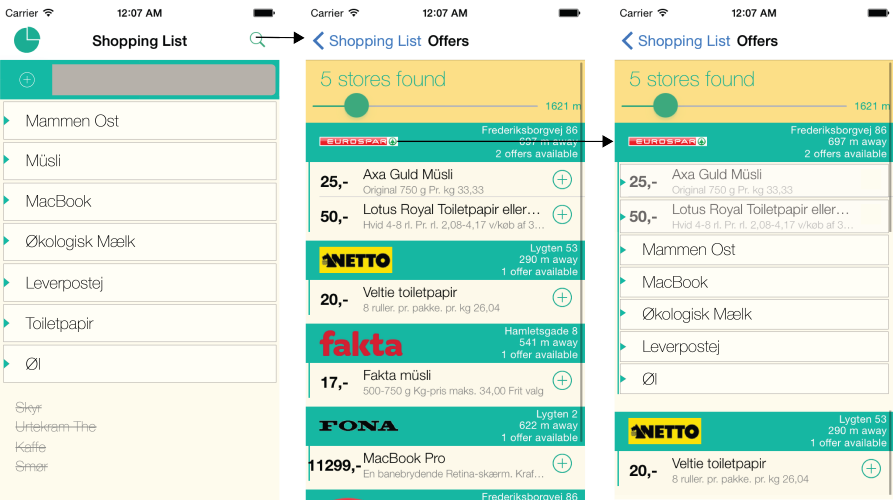


Fig. 5. Alternative prototype shopping list and store view design. This design makes it easier for users to understand why certain items are presented to them, but compromises their ability to quickly get an overview of where they can be bought.

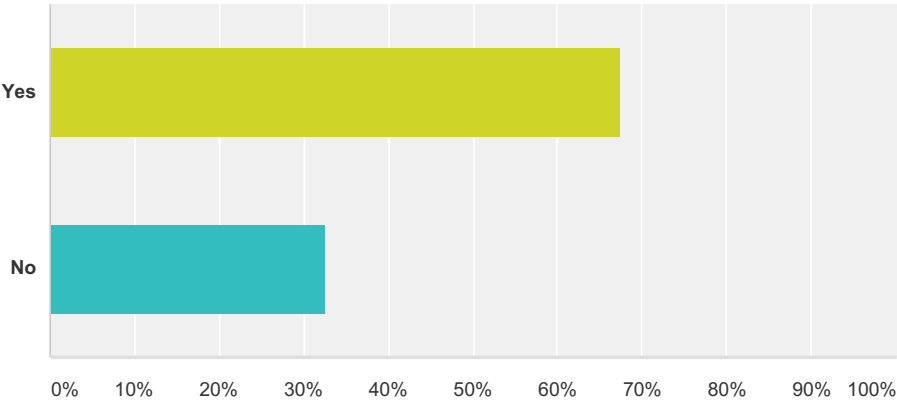
C Alternative design for prototype 3



D Survey

Q1 Do you shop for specific items on sale?

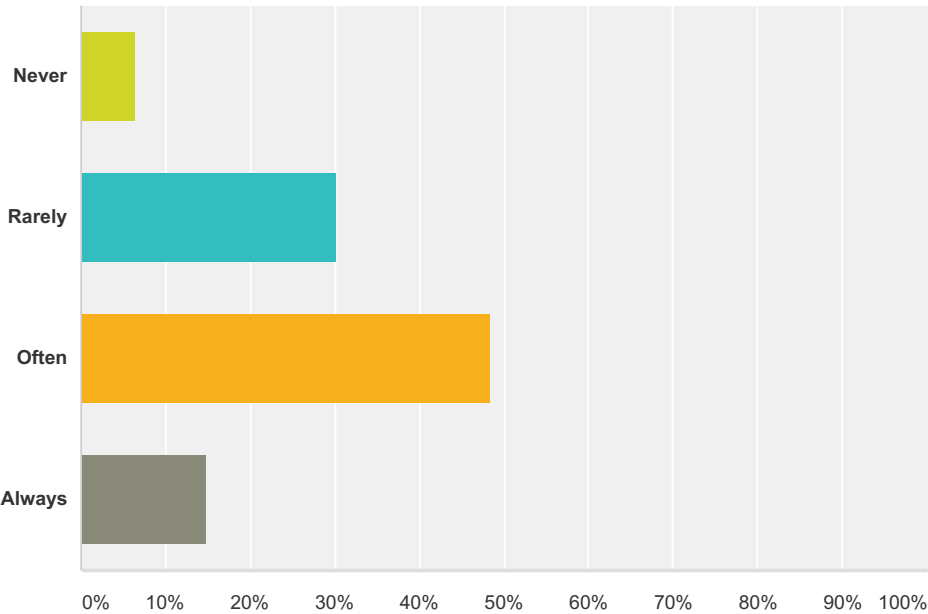
Answered: 157 Skipped: 0



Answer Choices	Responses	
Yes	67.52%	106
No	32.48%	51
Total		157

Q2 Do you look for offers when you are shopping?

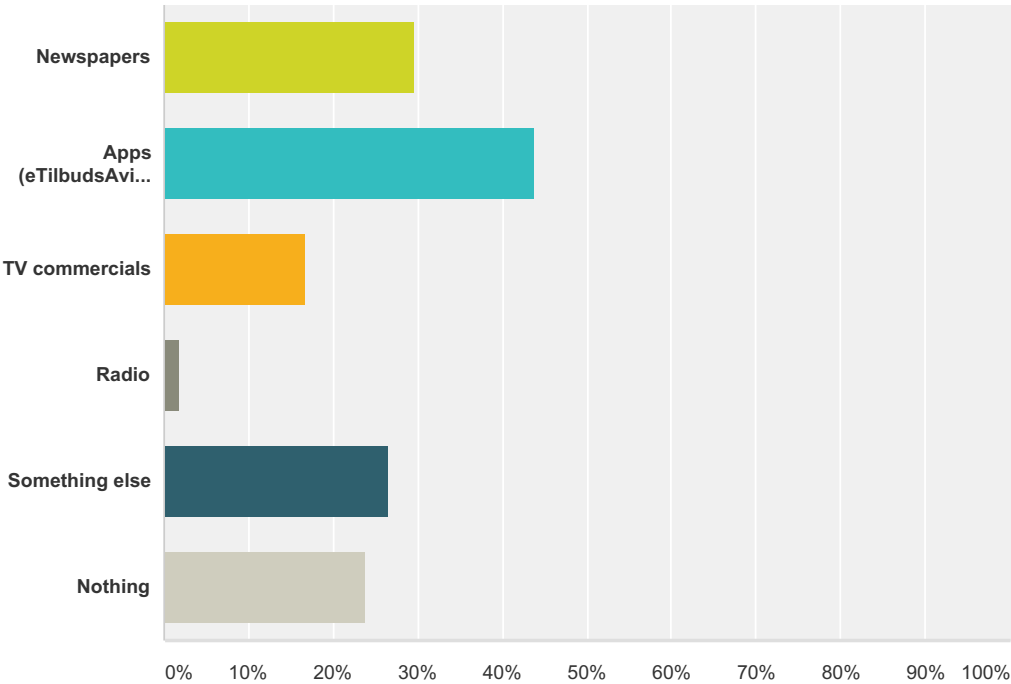
Answered: 155 Skipped: 2



Answer Choices	Responses
Never	6.45%10
Rarely	30.32%47
Often	48.39%75
Always	14.84%23
Total	155

Q3 What medias are you using to find special offers? (Possibility of multiple answers)

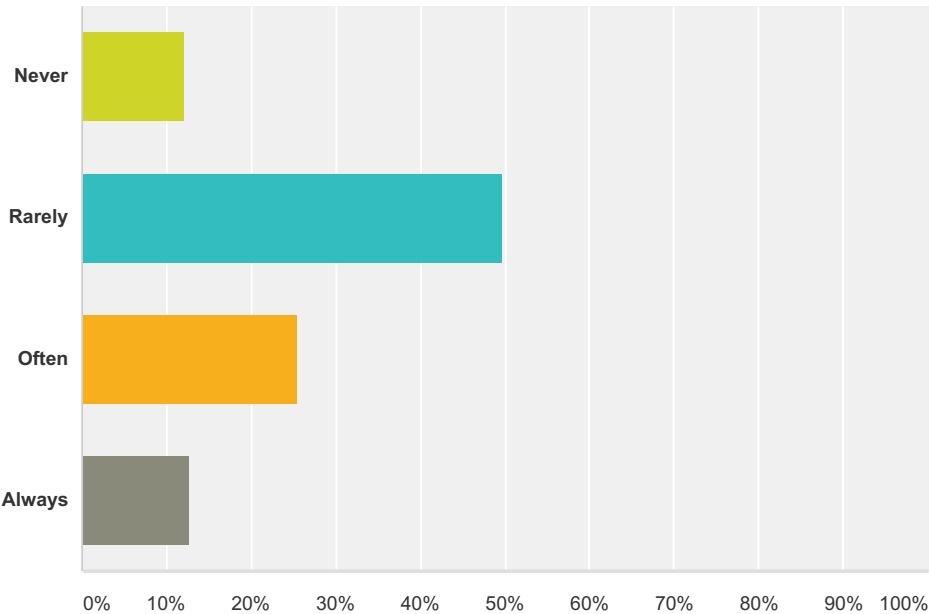
Answered: 155 Skipped: 2



Answer Choices	Responses	
Newspapers	29.68%	46
Apps (eTilbudsAvis etc.)	43.87%	68
TV commercials	16.77%	26
Radio	1.94%	3
Something else	26.45%	41
Nothing	23.87%	37
Total Respondents: 155		

Q4 How often do you make a shopping list?

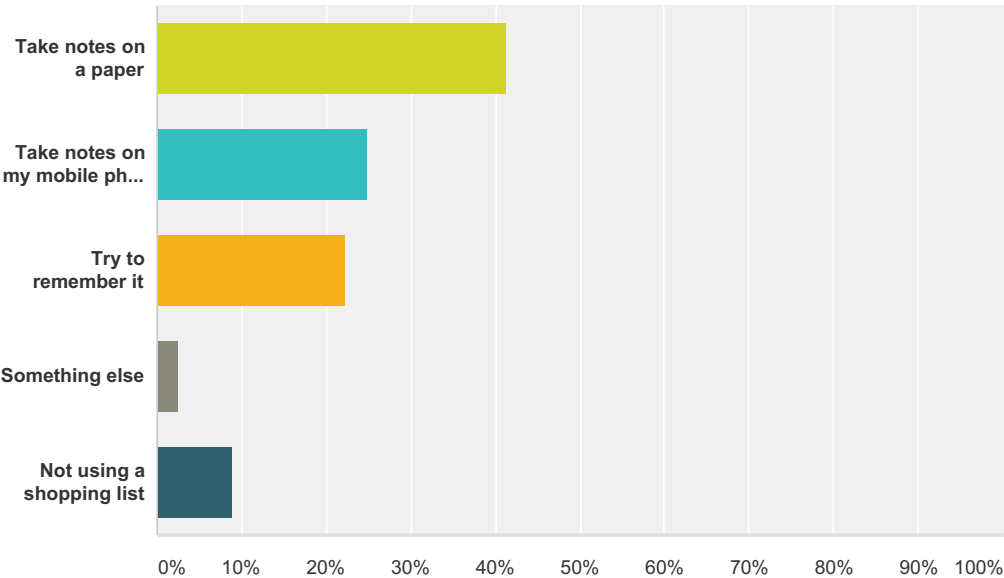
Answered: 157 Skipped: 0



Answer Choices	Responses	
Never	12.10%	19
Rarely	49.68%	78
Often	25.48%	40
Always	12.74%	20
Total		157

Q5 How do you keep track of your shopping list?

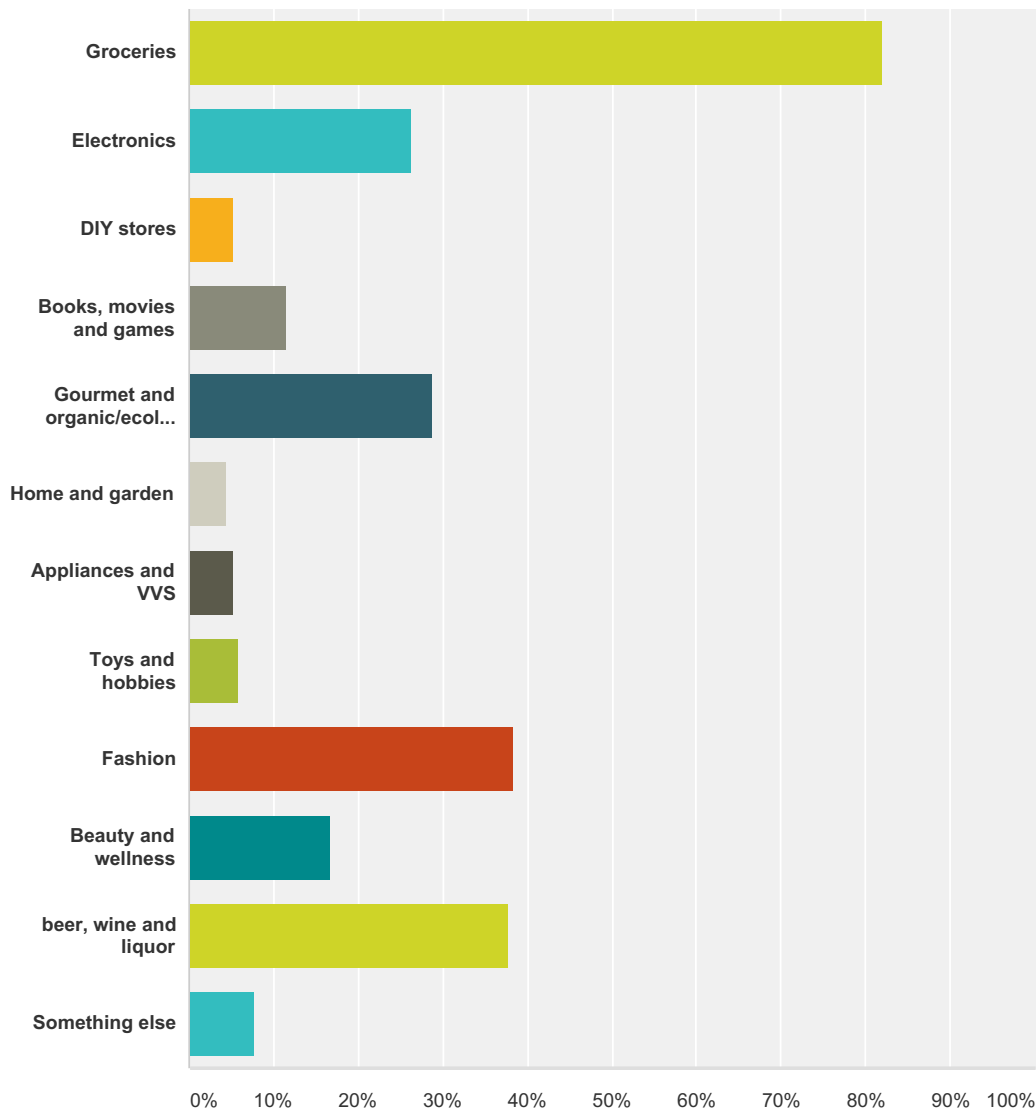
Answered: 157 Skipped: 0



Answer Choices	Responses	
Take notes on a paper	41.40%	65
Take notes on my mobile phone in an app	24.84%	39
Try to remember it	22.29%	35
Something else	2.55%	4
Not using a shopping list	8.92%	14
Total		157

Q6 What special offers are you mostly looking for? (Possibility of multiple answers)

Answered: 156 Skipped: 1



Answer Choices	Responses	
Groceries	82.05%	128
Electronics	26.28%	41
DIY stores	5.13%	8
Books, movies and games	11.54%	18
Gourmet and organic/ecological	28.85%	45
Home and garden	4.49%	7

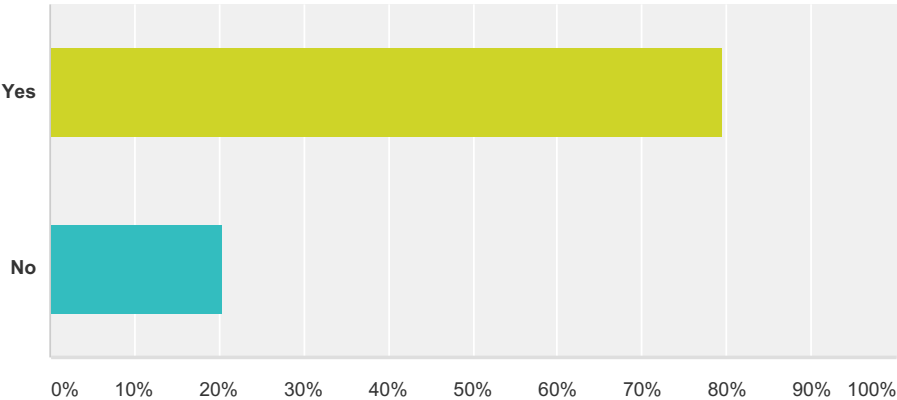
Mobile app feedback

SurveyMonkey

Appliances and VVS	5.13%	8
Toys and hobbies	5.77%	9
Fashion	38.46%	60
Beauty and wellness	16.67%	26
beer, wine and liquor	37.82%	59
Something else	7.69%	12
Total Respondents: 156		

Q7 Would you use a mobile app that can find offers near you that you are interested in and show it in an easy and straightforward manner?

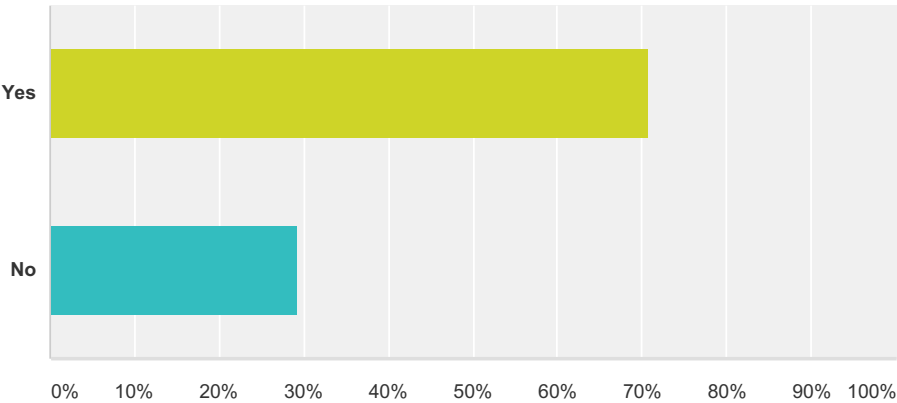
Answered: 156 Skipped: 1



Answer Choices	Responses	
Yes	79.49%	124
No	20.51%	32
Total		156

Q8 Would you be interested in getting an overview of your shopping habits?

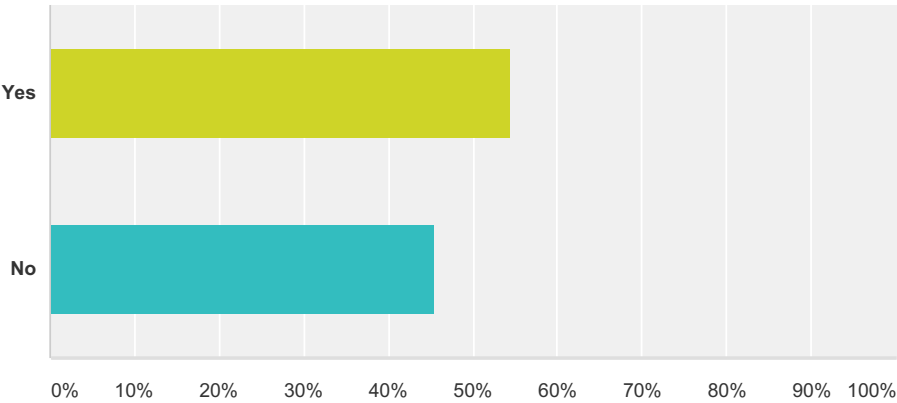
Answered: 157 Skipped: 0



Answer Choices	Responses	
Yes	70.70%	111
No	29.30%	46
Total		157

Q9 Would you be interested in receiving notifications showing offers that are relevant for you?

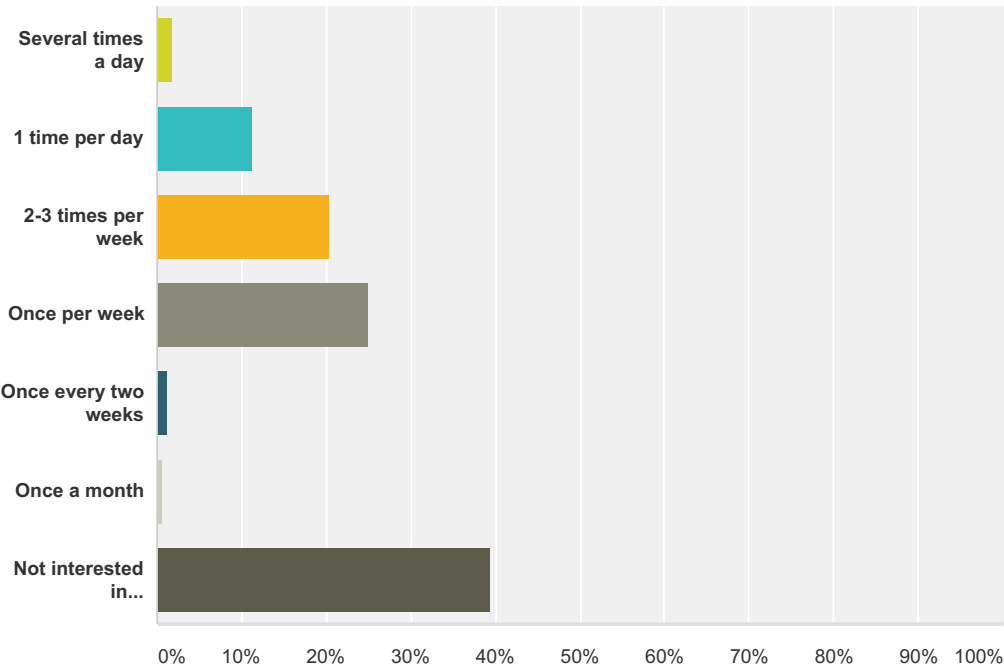
Answered: 156 Skipped: 1



Answer Choices	Responses	
Yes	54.49%	85
No	45.51%	71
Total		156

Q10 In that case, how often?

Answered: 152 Skipped: 5



Answer Choices	Responses	
Several times a day	1.97%	3
1 time per day	11.18%	17
2-3 times per week	20.39%	31
Once per week	25.00%	38
Once every two weeks	1.32%	2
Once a month	0.66%	1
Not interested in notifications	39.47%	60
Total		152