

Smart Shopping List

Abhinandan Deshpande¹, Deepak Ravindra Patil², Kanchan Bisht³ and Prashant Nagdev⁴

Abstract—How many times have you forgotten to buy something from the supermarket, that you had planned, and realized it after reaching home? It is more common than you think. It is in human nature to forget things. Unlike other items, groceries and daily use items are required to be bought again and again. But, the time gap between the two purchases of each item depends on the item, the consumer and some other parameters. We are trying to create a system which will track the usage of each item for each user and maintain a list of items which are needed to be bought again.

I. INTRODUCTION

The idea of the Smart Shopping List is highly motivated by the increasing stress levels in general population, let alone the students. In such a stressful scenario, forgetfulness is not a very rare sight. A lot of people don't have the time to go to the supermarket everyday, so the visits are weekly or bi-weekly. And forgetting to buy something which is going to be over before the next visit cannot be afforded. Hence, we have come up with an idea to develop a software that tracks the user's requirements and creates a list of things to be bought after each week.

It can be argued that a list of items on the phone can also be re-used every time, but the most important focus of our application is, to have more accuracy and reduce redundancy at the same time. A list of items in the phone can be re-used, but it does not take into account the difference in cycles of each item. Every item in the list has a different cycle of re-purchase. A bottle of dish washing liquid lasts a month, while one carton of milk is good just for a week. So, the next list should have milk on it and not dish washing liquid. If the user keeps re-using the traditional list of items, the accuracy of it will be less, since a dish washing liquid will be bought four times more than it's requirement. Eventually, the user will know of this redundant item on the list, and will get rid of the dish washing liquid from the list. Here comes another problem. "Out of sight is out of mind", not having dish washing liquid on the list will imply that the user has sufficient quantity of it at home. If the user forgets to put it on the list after four weeks, there is a great chance of him forgetting to buy it, and will have to visit the supermarket again.

Having considered the above scenario, we have come-up with a plan to create a software that uses it's intelligence to determine the cycle for each item, based on the user provided data. One important feature that we would like to introduce is, learning user behavior. There can be a case where the dish washing liquid was used up in twenty days instead of a month. In that case, the system will have to adjust the weights, and take this scenario into account for the calculation of the next cycle.

There is a discussion going on in our group, regarding the parameters that affect an item's usage. For example, in the month of December, most of the users go for a vacation or are in the festive mood. Hence, in such a period, the use of dish washing liquid will be less if the user is off to a vacation. But once the user is back in January, his usage will be as usual. The software needs to take care of such scenarios. This is an example of just one parameter. There are many such parameters which determine the usage of every item, and they have different effect on different users. For example, just like in the above scenario, the use of dish washing liquid would increase every December, if the user hosts a feast every December. Hence, we need the software to learn the parameters and their effect on the user, for every item. In order to facilitate this, we will require the user to supply the software with feedback after each of his visit to the supermarket. This will allow the software to adjust its weights.

There are some smart shopping assistants currently in the market, but most of them predict the things that should be bought by the user based on what he has bought already. Those assistants give us more options from the similar category but do not consider repeating the previous order. Here, we are proposing a system that merges our approach onto the traditional shopping assistant for the daily-use items, to create a shopping list for the user.

II. SURVEY

In order to get a concrete proof, that there is a need for such an application, we conducted a survey. After carefully discussing which questions should be asked in the questionnaire, the team has come up with five questions which help to decide whether there really is a need for an application like this. Our first question was for the volunteers to enter their email ids. The reason for this being the first question was to get valid data. It has been found that people give random answers when they know that nobody knows who answered them. After ensuring that the data we get from the survey is valid, we now started asking questions that will justify the need of this system in today's world. The surveyed people

*This work is a part of Software Engineering project.

¹Abhinandan Deshpande is a MCS student in North Carolina State University, Raleigh, North Carolina. Email: aadeshp2@ncsu.edu

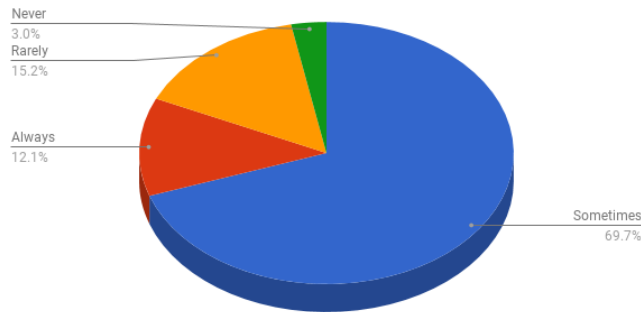
²Deepak Ravindra Patil is a MCS student in North Carolina State University, Raleigh, North Carolina. Email: dpatil@ncsu.edu

³Kanchan Bisht is a MCS student in North Carolina State University, Raleigh, North Carolina. Email: kbisht@ncsu.edu

⁴Prashant Nagdev is a MCS student in North Carolina State University, Raleigh, North Carolina. Email: psnagdev@ncsu.edu

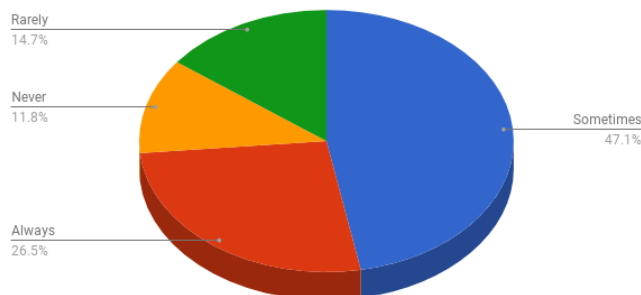
are the students of North Carolina State University. Most of these students are international students who don't have the resources to go for shopping regularly. Hence, they are our target audience.

A. Have you ever gone to buy items from a store but forgotten to purchase one (or more) of those?



First, we wanted to confirm that we are not the only people who forget to purchase things which we plan to buy. Hence, we asked the people whether they have ever forgotten to buy something from the supermarket, which they had planned to buy earlier. The response to this question tells us that most of the people forget things if not reminded about it. About 97 percent of the respondents tend to forget things. This makes the case for a need of a reminder to people.

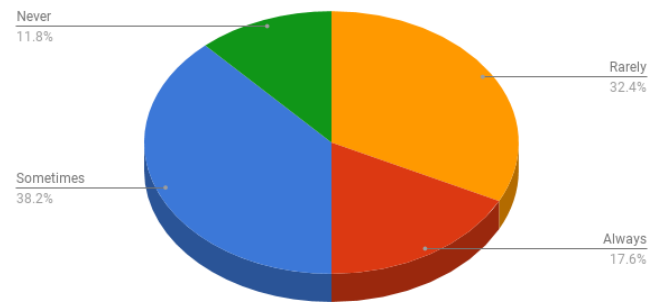
B. How often do you make a list of items to be purchased before visiting a store?



After confirming that there is a need for a reminder system for shoppers to know what things to buy, we asked the respondents whether they use some traditional method to remember the items to be purchased. This tells us whether people are making an effort to resolve this problem. Because, giving a solution to a problem for which no one cares to find a solution is of no use, the suggested software will not be used. Hence, we asked them whether they make a list of items before going to the store. As seen in the Fig.2, around 75 percent people have tried to tackle this problem themselves by making lists before going to the supermarket. From this response we can readily assume that, if we create a list for people to use, it will be received well, since more

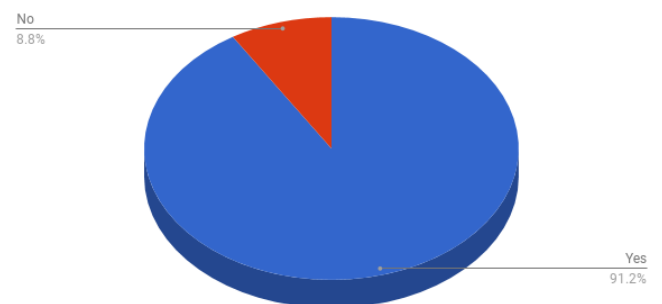
than 50 percent of the people try it themselves. Making this task of creating a list easier is, what we aim for. In order to do that, we first needed to know the loophole in the current way of making lists.

C. Do you ever forget to put items on your list(if you make one)?



Having known one of the major loopholes in the idea of making lists, we asked people whether they have ever forgotten to put things on their list. The response to this answer gave us the obvious way to proceed to the problem. From Fig. 3, we can see that more than half of the population forgets to put some items on the list. This makes it clear that the root problem is to remember what is to be bought. People usually tend to forget or are too lazy to check for everything. Hence, we figured that this was the main thing that we needed to fix. People need a software to remind them what things to buy, based on their previous behavior and shopping history.

D. Would you like someone to remind you what you need to buy and when?



Finally, we asked everyone whether they would like to have such an application which will remind them of the things to buy. And the result of the response has been unequivocal. More than 90 percent of the people have voted yes for implementing such an application of computer science. This kind of response led the way for us to lock this as our project for the subject. Now, moving forward from this survey, we needed some more data on the existing work done on this subject.

III. LITERATURE REVIEW

The magnanimous increase in the number of people having access to the internet and the increase in the handheld devices [6] has paved the way for a number of applications targeted for these users. Many of such applications exploit the basic everyday needs of a user. The more easy-to-use an application is along with being useful, the more it is expected to be used.

We have gone through the documentation of various patents that assist a consumer with grocery shopping in similar, yet, different ways. The following is the synopsis of these methods/ways:

The documentation of the patent Predictive Shopping notifications[1] briefly enlightens us about the flow of predictive analysis of a list of groceries and items that need to be bought for a consumer. The application analyzes the shopping history of consumers, based on which a purchase pattern for the corresponding consumers is generated. List generated by the application is used to remind the consumers about the products they need to purchase in near future. The list generated by the algorithm is also provided to the potential grocery sellers. If a discount is available on any of the products on the list, the customer is notified of this offer via mobile notification. Due to this smart technique, the customer ends up spending a minimal amount on a particular item. Also, the techniques may remind the consumer to purchase items on the proposed list on sensing that an item may be about to run out. This will, in turn, help the consumer from avoiding the situation of empty stock of important everyday items.

Study of the documentation of another patent, Shopping pattern recognition [2] discusses the technique of data mining to generate a list of items from customers history stored in retailers database. The techniques focus on the fact that the purchase patterns are consumer-specific and along with it, are product-specific, as well as, store-specific. For example, a consumer might buy fruits from Walmart but might buy cornflakes from the local convenience store. The shopping pattern of the consumer is determined by mining the transaction history of the consumer. The list of items that need to be bought in the future is generated based on the past purchases and the respective frequency of each product in the available list data obtained by data mining. The list is also sent to the retailers as a recommendation list. This allows retailers to help customers in shopping different items from their store, maximizing the profit of retailers. Customers save time by not making a list of necessary items and by avoiding extra trips to the shops/stores.

The technique discussed in 'Electronic shopping system and service' [3] makes use of computer servers that are equipped with predictive data collectors that are designed to collect the purchase predictive data of the user, interval of the purchase of the item(s) and frequency of the purchase. There exists a product recommender whose job is to predict the list of products that the user can buy along with the predicted items on his list. This product recommender is implemented

with the help of predictive data collector that recommends products similar to what the user has bought in the past. It may even recommend products based on what other users have bought in the past or are inclined to buy in the near future. Product recommender analyzes the consumer's data whether the consumer accepts the recommendation offered regarding any product or not. This information is used to increase its recommendation quality. This is useful for wholesalers, retailers, and consumer packaged goods (CPG) companies.

Another technique [4] utilizes a barcode scanner along with an intelligent base-station (any smart electronic device like a computer, smartphone etc.). The product barcodes are entered into the base-stations with the help of the barcode scanners. The base-stations have a linked database and are connected via the internet. The linked database belongs to a shopping service or a merchant. The database learns the shopping habits of the consumer over a period of time. The shopping list is created with the help of the data available in the database and is stored on the retailers server. The barcodes are downloaded and required information is gathered. This information is used to fetch the items from a store for a particular customer. Retailers server records the patterns of purchases made by users, and analyses these patterns to learn about the shopping habits of customers. Another advantage of this technique is that customer do not need to search for required items. Retailer suggests the items for the customer based on his past purchase history. This method can be used by the customer to shop via mail-order, telephone-order, in a store or via the internet using the created shopping list.

Another patent [5] describes the use of a budget to formulate a purchase list for a user based on his past receipts from different retail stores, pharmacies etc. It may also gather the user's online browsing information. It determines whether the items on the list can be purchased within the given budget. If true, the list is available to the user as is, otherwise, the items on the list are given a priority rating. Items with higher priority need to be accommodated on the list first, before making room for the ones with lower priority. After this list is available, the user has the option of including other items as well as removing any of the included items. If the user makes any changes to the list, the amendments are monitored by a machine learning engine that adjusts the priority rating of the concerned items, or the date of next shopping period, accordingly. This enables the system to be more accurate and more customer-specific with time.

In light of the summarized information from the various pieces of literature stated above, we conclude that the grocery products need to be purchased at regular intervals of time. The usage and purchase patterns of different users for the same product may vary depending on a number of factors. Some of these factors may be:

- Appetite of the user
- Affinity for the product
- Budget of the user
- Weather conditions (seasons)

- Routine at the time of concern (how busy the user is)
- Stress level (exams, feasts etc.)

IV. SYSTEM OVERVIEW

A. Implementation Plan

Our project mainly consists of two parts in which one is the client and another is the server. We are going to build an android application which will act as a client. The user will be allowed to enter details of purchases made by him. In order to achieve high accuracy, a large number of data consisting purchasing details is needed. Hence, it will be beneficial for the user if he enters data after every purchase. This will help our algorithm implemented at server side to generate a most accurate list of predicted items. Server for this application will be implemented using Nodejs and, call to the server will be made using a REST API. Backend server will have access to datasets that will be stored in the server-side database. MongoDB will be used for this purpose. By processing and analyzing the data, our recommendation engine will generate results for the next purchase cycle. When the user logs in for the first time, he will be asked to insert list of items. The user will get notification from the application at regular time intervals as per his purchase cycle. In addition to this, the user can access a list of items from the server anytime he wants.

will get more efficient predicted list. When the user enters data, it will be sent to the server and will be stored in NoSQL database. Recommendation engine which contains prediction algorithm will use this data to generate results which will be sent to the user. If the user wishes to erase the records of shopping, the application will ask for confirmation. If he confirms, his records (stored on the server) will be reset. The application will act as a medium to get the input of data in the form of a list of items and to display the predicted list of items. All the data will be analyzed on the server side and analyzed data will be used to display an output on the android application.

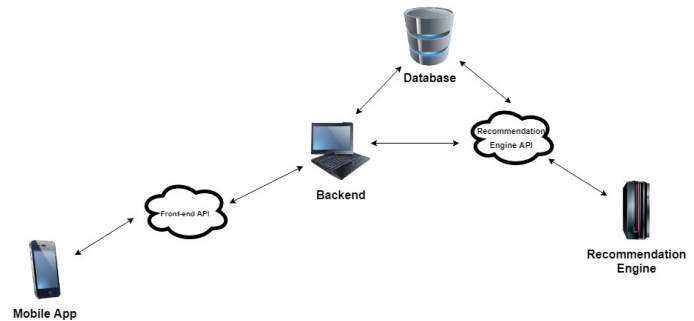


Fig. 2. System architecture

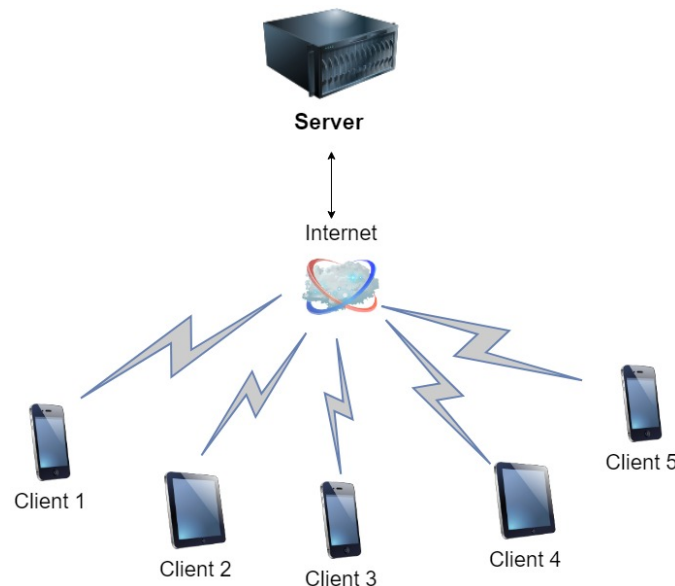


Fig. 1. Client server architecture

B. System Flow

First, the user will sign up for our application. After he signs up, initially he has to add his/her purchase data so that our algorithm can predict the items that need to be bought in the next cycle. After adding multiple purchase history, the application will be able to predict list for next purchase. The user will be able to sign in any time and he can get a list of items to be bought. As the user adds items frequently, he

C. Development Life-Cycle

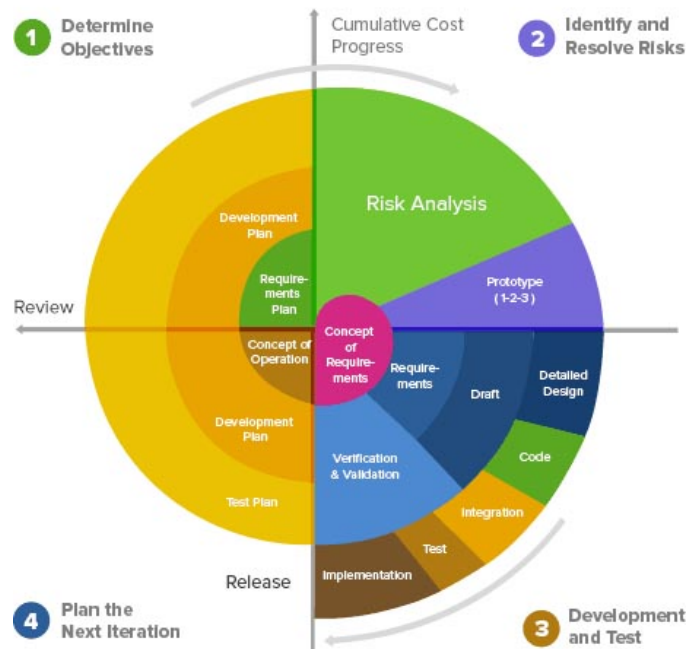


Fig. 3. Development process. Image source: <https://onlinedigitaltechnology.com/software-development-risk-aspects-and-success-frequency-on-spiral-model/spiral-model12/>

1) *Requirement Gathering*: The first thing that we planned to do was to get the overall idea of user re-

quirements. Hence, we conducted a survey which involved multiple questions. Each of the questions was designed to get a general idea of the application. According to the conducted survey, we found that most of the users are interested in the idea of such a reminder application. If the items missed by the user are essential, the user ends up making another trip to the grocery store. This trip wastes his money and time. Hence most of the users agreed to the idea of the application.

2) *Android Application Development*: In this stage, we will be focusing on making an efficient and easy-to-use android application which will allow users to send the data to the server. The data processed on the server will be sent back to users via this android application. We will be using the android studio for developing an android application.

3) *Server Development*: This is the second stage of our development. We will use NodeJs to implement the server-side code of our system. All the algorithms to process the data will be on the server. These algorithms will process the data, and, processed data will be stored in the database. We will use MongoDB as a database. The reason for using a MongoDB database is, it is a NoSQL database. And since, we are not aware of the data and its format beforehand, we would like to use MongoDB for this purpose. We will be trying out different algorithms to get higher accuracy and lower false alarms. We will evaluate those algorithms and choose the best one.

4) *Risk Analysis*: After implementing every algorithm, we will analyze the risks involved with that algorithm. Every algorithm would have some advantages and disadvantages. We would need to choose the one whose disadvantages are not the bottleneck for our application. For example, in this application, having false alarms is a bad sign but having high value of "False Negative" is worse. Hence, we will have to select an algorithm which satisfies this condition.

Along with the risks involved in the algorithms, there are some risks involved in the infrastructure, viz., security of data, data storage and authentication of users. We will focus on these risks as well.

V. EVALUATION OF SOFTWARE

Evaluation of any software is necessary to determine if that software works correctly. We will be doing the evaluation of our software by analyzing the list of items predicted by our android application. First, we manually create a predicted list of items (if the dataset is small). We will use our own bills to feed up the data to the application, and, check if the application predicts the correct list of items by comparing it with the manually created list of items. The application will be tested by four sample data sets and few other datasets. Based on the result of all these datasets, we will determine the correctness and accuracy of our smart shopping list. Evaluation will also be done on the below parameters:

- **Accuracy & Precision**: We aim to get higher accuracy and lower false alarm.
- **Scalability**: Our application should handle increasing number of users and provide a smooth service to each and every user using it. We will be using NodeJs for

this purpose. Since, it can concurrently handle many requests on the server-side.

- **Reliability**: Once the data is stored on the server, the user can fetch the data anytime he wants. If the user deletes the application accidentally, he shouldnt lose his data. Once he reinstalls the application, he can request for his data and his smart shopping list from the server. The data about the user will be maintained along with his username. Hence, whenever the user logs in with his credentials, he should be able to see the smart shopping list.
- **Security**: We will be using user authentication in order to allow access to user's data. One user will not be able to see the data items of other users.
- **Privacy**: The data of the user's grocery usage should be secure and private to him. The data should not be left unsecured and publicly accessible.

VI. CONCLUSIONS

The past experiences of our friends, relatives, acquaintances, as well as, that of our own selves with regard to grocery shopping, lead us to the idea of developing smart shopping assistant, in the form of a smart shopping list. After thoughtful consideration, we conducted a survey to gain insights into the usability of smart shopping assistant application. Results have confirmed the need for such an assistant that helps in remembering the shopping list. Our product would also help the users economically by helping them avoid extra trips to the market/stores.

ACKNOWLEDGMENT

We express our sincere gratitude towards our Software Engineering professor, Dr. Timothy Menzies, for giving us the opportunity to freely choose project idea, implement and test it using the tools and technologies of our choice. We also thank our fellow peers for helping us in requirement gathering by participating in the survey made by us. We thank the almighty for enabling us to choose this brilliant idea.

REFERENCES

- [1] Predictive shopping notifications US 8538807 B2 <https://www.google.com/patents/US8538807>
- [2] Shopping pattern recognition US 20160063511 A1 <https://patents.google.com/patent/US20160063511A1/en>
- [3] Electronic shopping system and service US 20140279208 A1 <https://www.google.com/patents/US20140279208>
- [4] System and method for a complete and convenient shopping experience US 7599855 B2 <https://www.google.com/patents/US7599855>
- [5] Predictive shopping WO 2017156067 A1 <https://www.google.com/patents/WO2017156067A1>
- [6] Rachel Bellamy, Calvin Swart, Wendy A. Kellogg, John Richards, and Jonathan Brezin IBM T. J. Watson Research Center, 'Designing an E-Grocery Application for a Palm Computer: Usability and Interface Issues', IEEE Personal Communications (Volume: 8, Issue: 4, Aug 2001, pg 60-64)