Tailoring Product Recommendations for Mobile Devices

Submitted in completion of the research project requirement of the CS848 W22 course in University of Waterloo

JUSTIN SAN JUAN and OWEN CHAMBERS, University of Waterloo, Canada

Since the beginning of the COVID-19 pandemic, online shopping has grown at a rate not seen before. Because of this, it is pertinent to display product recommendations in a way that makes it easy for users to understand what they are looking at and find what they need. Currently, there exists a lack of research on how to tailor the display of these recommendations to different devices. This is especially important for mobile devices where there is significantly less space to display all of the necessary information. In this paper, we explore two factors that influence a user's product selection decision process on mobile devices, mainly the size of the picture and amount of explanation for each product, and perform a user study to quantify the amount of influence these factors have.

Additional Key Words and Phrases: User Study, HCI, Product Recommendations

1 Introduction

Since the development of the internet into what it is to day, it has become impossible to use the internet without using a user interface. User interfaces are a point of human-computer interaction and communication [3]. These include everything from the mouse used to click on a screen, to the website being viewed and clicked on. A good interface allows users to find what they are looking for in as little time as necessary, while a bad interface can lead to wasted time, frustration, and a negative attitude towards the interface or website itself. One example where a good user interface is important is when giving product recommendations. For companies such as Amazon, it is important to provide users with the best items suited for their search, but also in the most appealing and straightforward format. If a user has a difficult time finding an item they are looking for, it may drive them to Amazon's competition. Because of this, Amazon focuses on making their interface as simple and easy to understand as possible even if it means sacrificing some information [10]. Amazon also includes measures of explainability along with the recommendations. Although Amazon may not elude to the exact reason the products were returned in the order that they were, they do include explanations for each of the items even if this information is negative such as a negative review. Amazon believes that even if the user does not buy the item, a more informed user will build more trust and be more likely to return in the future [11]. Although Amazon has found a format that is easy to use and brings repeat customers, there has been a lack of research into the best way to display product information on a mobile phone. In this study we make the following contributions: (1) We will outline two factors that influence mobile product recommendations including the size of pictures and amount of text, (2) build a prototype application, (3) perform a user study to test our hypotheses, and (4) evaluate the effect that these factors have on a users ability to select a product.

2 Background and Motivation

Although it is easy for Amazon to provide all of the necessary information on a desktop device, it soon becomes a problem of prioritizing information when the screen size is restricted, such as with a mobile phone or tablet. These devices come with their own set of technology- and user-related problems [14]. User-related issues include aspects such as users changing locations, contexts, and personal preferences while technology-related issues include limited battery life, fewer buttons, and a smaller display. Because of the smaller screen, it is impossible to display products on mobile

Authors' address: Justin San Juan, justin.sanjuan@uwaterloo.ca; Owen Chambers, ochamber@uwaterloo.ca, University of Waterloo, Waterloo, Ontario, Canada.

Hypothesis	Description
1	Less text and larger images will allow users to accomplish the task in less time and with less clicks.
2	Smaller pictures and more text will make the users more confident in their decisions
3	Larger images will be better to display items more efficiently and lead to a greater intention to re-use

Table 1. Hypotheses

the same way as on desktop. Mobile recommendations must cater to the limited screen. Although it is understood that this needs to be addressed, there exists a lack of research on how changing the size of different parts of a recommended item can influence a person's ability to find the item that they like the most.

In this study, we attempt to identify the optimal relationship between the size of the product image and amount of explanation presented when searching for a product. This is important on mobile devices as a smaller image allows for more explanation, but the extra information may not be required for this medium, or may even be detrimental. To test this, we created five different user interfaces that change the size these variables to identify which of these factor leads the the optimal product representation. These interfaces include a small image with short text, a large image with short text, a small image and long text, and a large image with long text. The last interface has a medium sized picture and text used as a controlled interface when evaluating results. Aside from how the results are being displayed, it is also important to consider what is being searched as many items have different requirements that users are looking for. This can be seen when imagining people searching for two different items: a refrigerator and a drone. When shopping for a refrigerator, most people are concerned that it is the correct size but may not have firm opinions on the brand, lifetime, and other aspects. This is in contrast to someone who is looking to purchase a drone. When doing so, a user will look at the brand, maximum flight duration, maximum distance away from receiver, noise, and other aspects. Understanding the task at hand, it is important to ensure that the user study performed is being carried out in a manner that accommodates these differences.

2.1 Hypothesis

Through conducting an experiment, we have outlined three hypotheses that we believe can be answered as shown in Table 1:

Because refrigerators have a small amount of features that users look for compared to other items such as drones or electronics, less text and larger images will allow the user to accomplish the task more efficiently and with less clicks. Hypothesis 1 will be measured through quantitative measures including the amount of time and number of clicks used. By showing the interfaces in different orders, the larger pictures and more text should have less time spent and clicks.

Hypothesis 2 will be measured through two questions on the survey shown in Table 5. The first being "This interface did not make me confident in my decision" and "This interface lacked necessary information". Since the interfaces with more text will allow the users to be more informed, we believe it will allow them to be more confident in their decision. The second question will determine if this extra information is necessary or if the user's would have been just as content without the information. These are questions 2 and 5 on the questionnaire.

Hypothesis 3 will be measured by two different questions on the survey. They are "This interface displayed information efficiently" and "I would want to use this interface again in the future". Because hypothetically less information is required when purchasing a refrigerator compared to a drone, the most efficient way to display the information will be

with large pictures. Since we believe the larger pictures will allow the users to select their preferred items faster, this will lead to them wanting to re-use the interfaces with larger pictures.

We have also included two extra questions to gain extra insight on the user's thought process including, "This interface provides quality results" and a free form question where users are asked to input the three features they look for most while shopping for a refrigerator. By doing so, we hope to gain insight into how the users view each of the interfaces on a higher level.

3 Design

User studies are used in many fields in order to understand how a system will be used by a user, as well as the extent of which independent variables have effects [13]. User studies have been used many times when researching how users interact with interfaces as it can be an effective way of understanding how the user will use the system. Although it is easy to understand that a user study will be useful, it can be difficult to determine the best way of implementing it. The two main factors to consider include the variance in the test set up, as well as correctly and appropriately measuring the results. This section describes the design decisions made for the user study.

3.1 Minimizing Variance

Before the details of a user study can be decided upon, it is most important to choose the correct type of user study. This comes down to two options, between-subjects and within-subjects, each with their own pros and cons [1, 13]. Between-subject tests involve different people testing each condition, so that each person is only exposed to a single variation, while within-subject tests involve each user interacting with each variation. There are many pros and cons for each. Between subject tests are easier to set up since the order of interfaces shown do not have to be tracked, each participant only requires a shorter session since they only see one interface, and this method prevents "learning" which enables participants to complete tasks more efficiently as they repeat it and skews results. Within-subject tests have their own advantages to consider as well. Within subject tests work better when a lower number of participants are available since more information can be achieved from each one. This also reduces the amount of noise in the data in the form of differences in user preferences. In the case of between-subject tests, if one user is happy and the other is sad while taking the test, those emotions will influence the outcome of only the interfaces seen by those participants. With a within-subject design however, this emotional "noise" gets spread across each of the interfaces more evenly, thus distributing the skewing effect of these factors. For this task, the best type of study to use is the within-subject design because most of the benefits of the between subject study do not apply. The time for a user to complete the task and fill out the survey only takes several minutes with each of the interfaces, thus time is not an issue. As far as learning from the first task since the task is simple, straight forward, and familiar to the user, the information gained after the first task will not influence the user's ability to complete the task for the second and third variations to the point where it is a detriment to the study. Since within-subject studies provide more information with limited participants, this is the best form of study that can be performed.

As well as choosing the correct type of study, it is also important that the task performed represents that which the users would see in the real world [9]. We have taken many steps in order to insure that this is the case for our study. This is most notably done by creating consistency in the search results as discussed in this section.

Algorithm 1 Acquire the ordered subset of results for an interface

```
N ← total number of interfaces
T ← total number of items
s ← session unique ID
idx ← interfaceIndex
r ← array of items
Require: i ∈ {0, ..., N − 1}
targetIndex ← (hash(s) + idx)%N
results ← empty array
for i in {0,...,T-1} do                          ▷ A simple filter function
    if i % N == targetIndex then
        results.append(r[i])
    end if
end for
return results
```

3.2 Query Used and Cached Recommendations

The results provided by a company like Amazon vary wildly depending on the query selected. Because of this it is possible that someone who enters a good query has a much easier time selecting an item than someone who enters a bad query. This introduces variability that is not being measured in this study which could lead to invalid or skewed results. To combat this, we have created pre-formulated queries and cached the results so that each user sees the same results. To ensure the query resembled what the actual user would see is by selecting our chosen query to resemble that of which may be chosen by the user when searching for the item at hand, and provide a range of results since each participants tastes will differ and therefore be accounted for. In addition, the usage of saved of results reduces variance in waiting times through as the participants load the recommendations.

3.3 Preserving the Order of Recommendations

Since amazon returns products in a specific order with higher-ranked results being items that the website scores as more relevant, showing participants the products in the order presented stopping at a cut off point to switch interfaces would introduce unnecessary variability since the last interface would get the products deemed by Amazon as not as relevant. To address this, we take the products given by Amazon and distribute the results across interfaces in an alternating fashion described in Algorithm 1.

First, the total number of interfaces N, and total number of items T are obtained. Each user is given a unique session ID s, and the current interface that the user is testing is provided an index number idx. From these, the $targetIndex$ is calculated by hashing s, adding idx to it, then using the modulo operator on the result with N as the divisor.

Finally, the items in the input array r with index i returning the same value as $targetIndex$ when $i\%N$ is evaluated are collected as the $results$.

The algorithm accomplishes three objectives in conjunction. First, for each interface, it retains the order of recommendations, thus avoiding biased sampling. Then, across interfaces, it allows each product set to be equally likely to be shown for that interface. Finally, for each user, each product will only be shown once, thus preventing learning variance based on the product content.

3.4 Time-Constrained Task

The last factor we included to minimize variance is a time constraint. The influence of time constraints has been studied on many occasions including assessing the relevance of documents [12]. In this study, it was shown that applying a time constraint did not negatively affect a persons ability to assess the relevance of documents, however a shorter time constraint did make participants more stressed. Time constraints have also been used to study user centered evaluations of search interfaces [9]. In this case, researchers note that the time constraint better represents real world scenarios as users may take more time in a study than they would when searching on their own, but also note that it imposes a level of stress. Because we want our study to closely resemble the task posed in the real world we have imposed a minor time constraint to ensure that the users do not take an absurd amount of time but also do not get too stressed while completing the task.

3.5 Evaluating Results

The second part implementing a user study is ensuring that the results are being measured accurately. This includes making sure the results are being recorded in the right way and that what is being measured answers the hypothesis being evaluated. This can be done through quantitative and qualitative measurements. Qualitative data is descriptive and conceptual while quantitative data can be counted, measured, and expressed using numbers. Both methods have a deep history of being used in different contexts and come with their own pros and cons. Initially, quantitative studies were seen as producing legitimate answers where changes are able to take place while qualitative measures were used to discover knowledge to be tested [2]. This changed however as more advantages of qualitative measurements were seen and began being adopted more in the 21st century [5]. Mixed-method research was created to balance the pros and cons of each method to maximize the advantages and minimize the disadvantages of both methods [5].

Because of this, we have decided to take advantage of both qualitative and quantitative data. Quantitative data includes the time users spent using each of the interfaces and the amount of clicks used. This data is not influenced by the user's opinion and directly relates to how simple or convoluted each interface is. Qualitative data consists of a survey with various questions that answer the posed hypothesis questions. This gains insight into the preferences of the user that may not be reflected by the quantitative data. Many questionnaires exist, each with a different method of receiving feedback from the user.

One popular example of this is the Likert scale which asks a question and then asks the user to state the degree to which they agree with the statement with options such as strongly disagree, disagree, neutral, agree, and strongly agree. Giving options allows the user to state their opinion without the need for a wordy answer using a text box, and limits the number of choices they have since selecting from 10 options would be overwhelming. The Likert scale also includes answers from a spectrum of results including disagree and agree options instead of just agree which allows the user to voice any opinion they have. For this reason a Likert scale was chosen for this experiment. Because this experiment has never been performed, questions were then tailored to answer the hypothesis as well as gain insight from the user's perspective. These questions were asked in a neutral way that does not elicit answers from the user and covers each of the variables being researched.

3.6 Independent Variables

The independent variables in the experiment are described in Table 2. A small image size reduces the area of the image to 40% of the fillable image space. For medium and large, it is 70% and 100% respectively. It is specifically noted from a

Variable	Variation	Effect
Image Size	Small	Image area is 40% of image container
Image Size	Medium	Image area is 70% of image container
Image Size	Large	Image area is 100% of image container
Text Length	Short	Maximum of 2 lines
Text Length	Medium	Maximum of 4 lines
Text Length	Long	Maximum of 6 lines

Table 2. Independent variables tested and effect on the interface

Interface Name	Image Size	Text Length
Control	Medium	Medium
Small-Short	Small	Short
Small-Long	Small	Long
Large-Short	Large	Short
Large-Long	Large	Long

Table 3. Configurations for each interface tested

data visualization perspective that the area of the image is linearly being modified instead of each dimension (width and height) in multiplication. Images are fitted into the modified area using the "contain" mode, which indicates that the larger of the width or height is resized down to fit the modified area, and the original aspect ratio of the image is maintained. For text length, a short variation uses a maximum of 2 lines, while it is 4 and 6 for medium and long respectively. It is noted that some recommendations will not have a total of 6 lines, and thus the text length limitation may not create an effect.

The interfaces tested are then described in Table 3. The control interface uses the medium variation for both Image Size and Text Length. Then, the lower and higher variations are tested for each smaller / shorter and larger / longer variation of the independent variables. This results in four additional interfaces being Small-Short, Small-Long, Large-Short, and Large-Long. By using these interfaces instead of all of the 9 possible combinations (low, medium, high for each independent variable), a large coverage of variable effects can be observed and still meaningfully extracted during analysis.

3.7 Controlled Variables

In order to reduce the noise in the display, several factors about the recommendations are modified as disclosed in this section. The controlled variables include:

(1) Height of each recommendation
(2) Free shipping tag
(3) Price is always included
(4) Removed "sponsored", "prime" tags
(5) Undisclosed ratings

Each recommendation, regardless of large or small variations for the independent variables, have the same height. This reduces additional variance in number of items above the fold from having smaller images, shorter text, etc. White

space takes the place of shrunken images or shortened text. Meanwhile, "free shipping" tags are included for all items to provide consistency. Then, items that do not have a price shown due to having multiple options are replaced with the first item that has a price for the selection. Additionally, the recommendations often include tags such as "sponsored" and Amazon "prime". These tags are removed regardless of the item. Finally, the ratings are hidden from both the recommendation list as well as the inner product display. These changes are made to enable the user to focus on the impact of the independent variables tested in the experiment.

4 Implementation

4.1 Interface Creation

The prototype application is created using React Native. The project is demo-able using an Android device. While it is portable to iOS without code change, it has not been tested on Apple devices. Four types of screens are created: Home, Recommendations, Product, Complete, and Setup. These screens are shown in Figure 1.

(1) **Home** contains a search bar and a prompt to being searching.
(2) **Recommendations** contains the list of results returned by the interleaved slicing algorithm in Algorithm 1.
(3) **Product** shows more information about the item selected. It contains the title, image, list of product details, quantity select (not interactable), add-to-cart button, and buy-now button. Both add-to-cart and buy-now will simply increase the count of selected items and bring the user back to the Recommendations screen.
(4) **Complete** shows that the task is complete and the user can click the button to go to the survey form.
(5) **Setup** contains userId, taskId, interfaceId, sessionId configurations and buttons to reset the selected items count. It also contains buttons to open survey links. It is only accessible by performing a secret action and should not be viewed by the participant. To access the screen, the title in the Home or Complete screen must be pressed 6 times in 2 seconds.

The general flow of user interaction goes from Home to Recommendations to Product to Complete. The application is publicly accessible at https://github.com/justinsj/recommender-system-ui [8].

4.2 Collection of Items

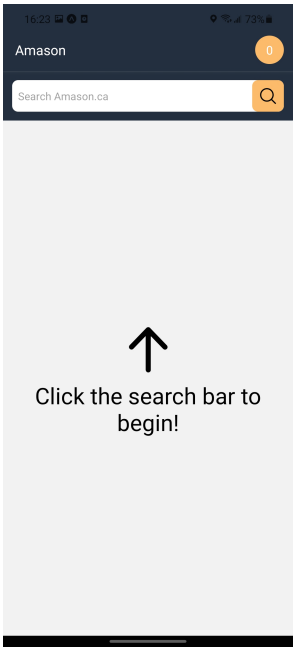
The results of queries on Amazon are collected using web scraping. A semi-automatic custom Python script using the Selenium webdriver is created to collect a total of 104 unique items recommended from the query "refrigerator". The web scraper is publicly accessible at https://github.com/justinsj/recommender-system-scraper [7].

4.3 Logging of Actions

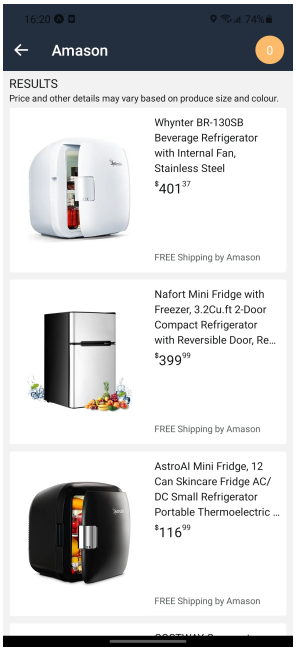
The actions of the users are logged through remote API calls. The implementation uses an AWS DynamoDB to store the results, with the API call going through API Gateway and triggering a simple Lambda function for input validation as shown in Figure 2. These logging actions are performed asynchronously and do not interfere with the user's interactions with the interface.

4.4 Logged Events

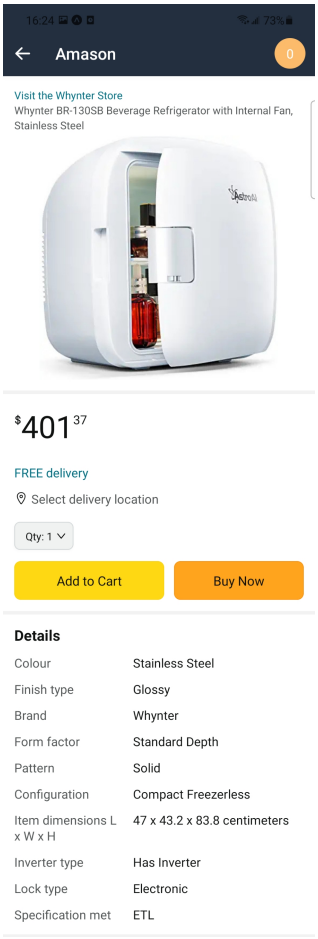
The events logged in the system are shown in Table 4. They include item views, clicks, and events of add-to-cart / buy-now. View events occur when the item enters the viewport of the user. Correspondingly, view reverses occur on item exit. Click events occur when the item is clicked into, thus entering the product information page. Click



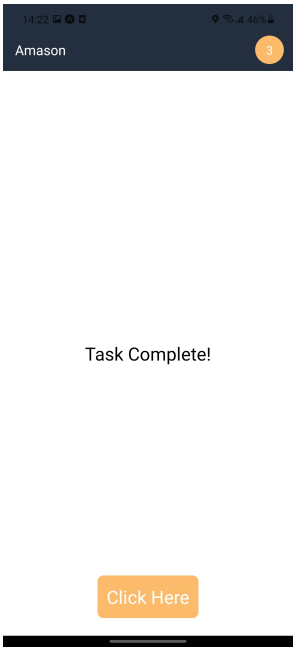
(a) Home screen



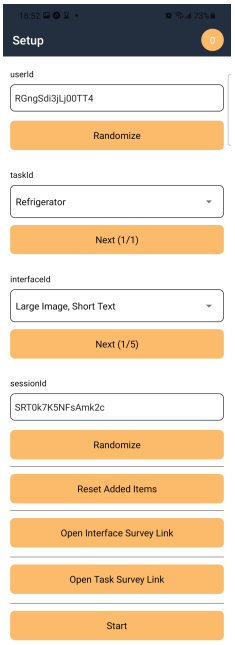
(b) Recommendations Screen



(c) Product screen



(d) Complete screen



(e) Setup screen

Fig. 1. Interface screens created

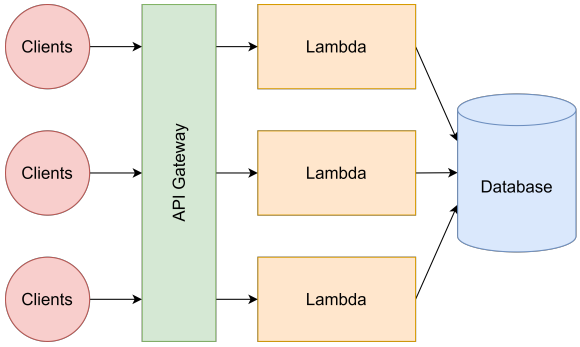


Fig. 2. Component architecture for logging actions

Name	Description
View	Occurs when product is in scroll view
View-Reverse	Occurs when product leaves scroll view
Click	Occurs when product is clicked into
Click-Reverse	Occurs when going back from the product screen
Add-To-Cart	Occurs when clicking the "Add to cart" button in the product screen
Buy-Now	Occurs when clicking the "Buy now" button in the product screen

Table 4. Names and descriptions of logged events

reverses occur when the user returns to the previous page (e.g. by clicking the back button on the header bar or after adding-to-cart / buying-now. The add-to-cart and buy-now are triggered when the corresponding button is pressed.

Each log consists of the following keys:

- **userId**: a unique ID
- **ts**: timestamp in ISO8601 format
- **taskId**: the task name (e.g. refrigerator, drone, etc.)
- **interfaceId**: the interface name (e.g. control, small_short, etc.)
- **sessionId**: another unique ID
- **productId**: a unique ID of the product
- **addedItemsCount**: number of items added-to-cart or bought-now
- **action**: any of view, view-reverse, click, click-reverse, add-to-cart, buy-now

4.5 Analysis of Actions

This section discusses the methods employed to analyse and measure the target metrics of task completion time and click through rates.

To measure task completion time, the logs are first grouped by unique userId-sessionId keys. These groups are then further grouped by taskId and interfaceId. Finally, the task completion time is calculated as the time between the first event (likely a view event) and the last event being the earliest event that includes in an addedItemsCount of the total required in the task (likely an add-to-cart or buy-now event).

Number	Question	Hypothesis
1	This interface displayed information efficiently	3
2	This interface did not make me confident in my decision	2
3	This interface returned quality results	Extra
4	I would want to use this interface again in the future	3
5	This interface lacked necessary information	2

Table 5. Survey Questions

Meanwhile, click-though-rates are measured by first creating similar groupings using the userId, sessionId, taskId, and interfaceId combinations. Then, the ratio of the set of items clicked (based on the productId) divided by the set of items viewed are calculated as the click-through-rate for the interface regardless of how many times they were viewed or clicked.

To reduce the effect of different averages for both task completion time and click-through-rates across users, the following method shown in Equation 1 is used. The average for each measure is calculated by first calculating the delta of different interfaces compared to the control for each userId-sessionId key. These deltas for each interface are then averaged across users.

It should be noted that ratios involving add-to-cart or buy-now are not metrics since they are in the task requirement to select a specific number of items.

$$\overline{m}_i = \frac{\sum_{j=1}^{J}(m_{i,j} - \overline{m}_j)}{J} \tag{1}$$

where:

m := is the metric being measured

I := total number of interfaces

i := interface index, $i \in \{0, ..., I\}$

J := total number of users

j := user index, $j \in \{0, ..., J\}$

The implementation of analysis is publicly accessible at https://github.com/justinsj/recommender-system-analysis [6].

4.6 Survey Questions

Table 5 lists the questions completed by the users after completing the task for each interface. Some of these questions are phrased negatively so as to not bias the person completing the task. These questions are answered on a scale from 1: Strongly Disagree to 5: Strongly Agree. At the end of the study users are asked to complete a 6th question that asks them to list the top three features they looked for while completing the task. The hypotheses that these questions pertain to are listed in Table 1.

5 Evaluation

5.1 How the Experiment is Executed

Understanding the requirements that make a good user study and with the created interfaces, we discuss how this user study will be implemented. The protocol of the user study is described in Algorithm 2. At the start of the user study, the

Algorithm 2 The protocol for performing the user study

```
u ← user
p ← proctor
N ← number of items to add
T ← totalnumberoftasks
I ← total number of interfaces
device ← mobile device with app

tasks ← list of tasks
randomTasks ← getRandomOrder(tasks, u)          ▷ The user's ID is used as the seed

interfaces ← list of interfaces
randomInterfaces ← getRandomOrder(interfaces, u)          ▷ The user's ID is used as the seed

for t in {0,...,T-1} do
    task ← randomTasks[t]
    p.setTask(device, task)
    p.sayTo(u, "Select your top N {task.name} items that they would consider buying in 5 minutes.")
    for i in {0,...,I-1} do
        interface ← randomInterfaces[i]
        p.setInterface(device, interface)
        u.testOn(device)
        u.openInterfaceSurvey(device,task,interface)
        u.fillInterfaceSurvey(device)
    end for
    p.openTaskSurvey(device, task)
    u.fillTaskSurvey(device)
end for

p.thankForParticipation(u)                      ▷ User study is complete
```

user is explained the task. The task being that they are going to be shown five interfaces. For each interface their goal is to select the three products that they would be most interested in purchasing in a real world situation in under 5 minutes. This should be plenty of time to complete the task but just informs the user not to take an exorbitant amount of time. The user is then presented with the first interface such that each interface is shown in a random order for as equal an amount of times as possible, such as in a Latin Square design. For example, small image with short text should be shown first, second, third, and fourth the same amount of times along with every other variation. For each interface, the user selects three products and informs us when they are done. The software keeps track of how long it takes the user to do this as well as the number of times they click back and forth on each product. After they have completed the task for the first interface, they are asked to answer the questionnaire for that particular interface consisting of 5 questions listed in Table 5. This is done so that the interface is at the forefront of the user's memory and it cannot be confused with the other interfaces. This is then repeated for the rest of the interfaces so the user will answer each of the questions for each interface. After doing so, the user is thanked for their participation and the responses are anonymously saved.

		Image Size		
		small <–> large		
Text length	short <–> long	+2.39s	←	+8.61s
		↑	0.00	↑
		+14.23	←	+29.30s

(a) Task completion times for each interface compared to control

		Image Size		
		small <–> large		
Text length	short <–> long	-1.10%	→	+5.85%
		↓	0.00	↑
		+2.02%	←	-1.00%

(b) Click-through-rates for each interface compared to control

Table 6. Results of preliminary testing

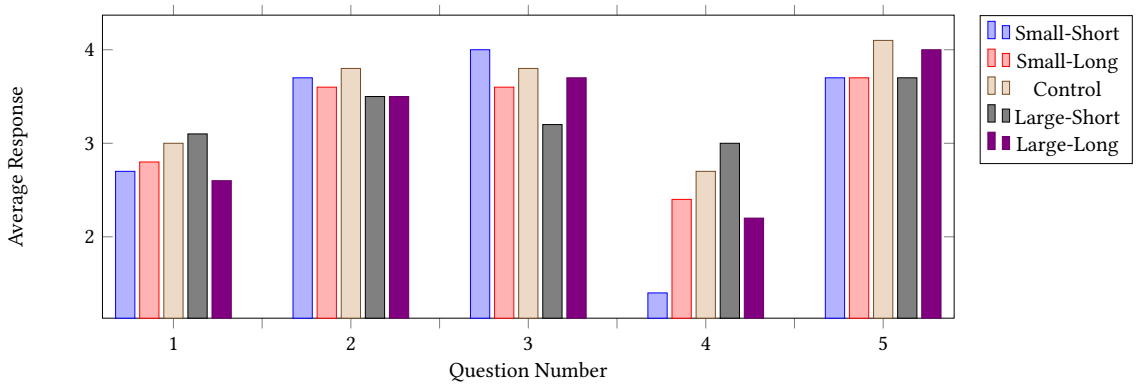


Fig. 3. Results of interface survey questions answered after each interface.

6 Results

The results of the experiment with 10 participants are shown in Table 6. It is noted that these results are not statistically significant. Table 6a shows that the control has the lowest task completion time, wherein all other interfaces have a positive ΔTCT. It can be noted that for both short and long text, a smaller image shows shorter task completion time, while for both small and large images, longer text shows shorter task completion time. These favourable changes in the metric are indicated by the arrows in the table.

Table 6b shows the change in click-through-rates of products for all of the interfaces tested. It is shown that long text with a large image increases click-through-rate, while a shorter text with smaller images also increase click-through-rate. The favourable direction of length of text or size of image is not clear from the results. Approximately 600 participants maintaining similar results are needed to show a 95% confidence statistically significant result in the effect observed.

The average click-through-rate in the experiment is 21.14%, and the average task completion time is 51.47s. These preliminary results oppose our Hypothesis 1, and suggest that longer text and smaller images may correlate with shorter task completion time with decreased click-through rate.

Figure 3 shows the results from the survey questions asked in Table 5 after performing the task for each interface. The average response for each question is calculated and displayed. It should be noted that given the small number of participants in the pilot study, the results are not statistically significant. These following observations then only provide sample insights from the designed study. Our second hypothesis that smaller pictures and more text will make

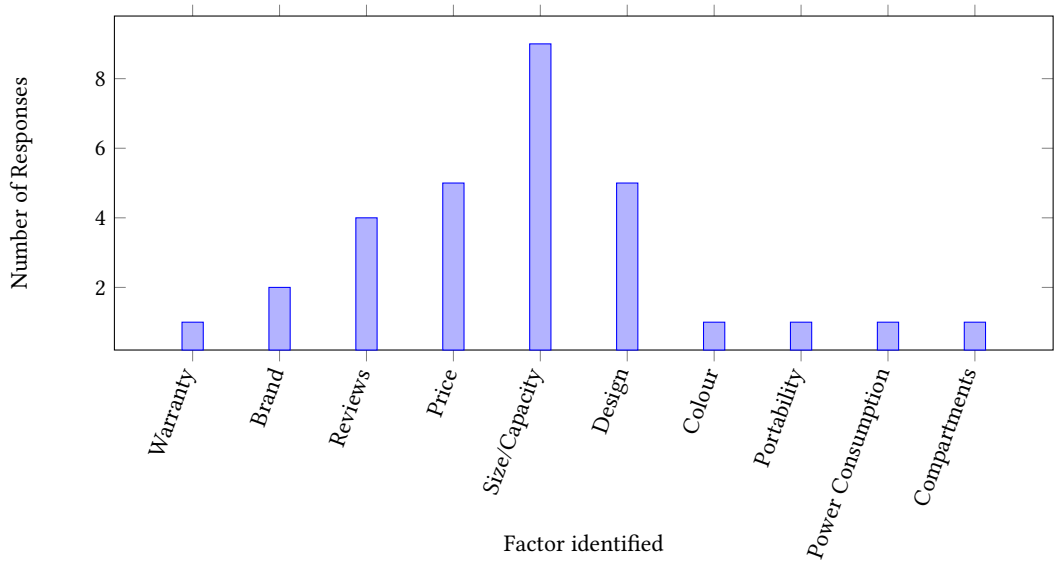


Fig. 4. These are the results users identified of basing their decision on the most.

users more confident in their decision was measured by questions 2 and 5. It is important to note that these questions were phrased negatively which means a lower score for questions 2 and 5 relate to more confidence and more necessary information respectively. Interestingly, the control interface made the users the least confident. We can also see that although the values are close, longer text and larger images made the users more confident. Question 5 shows similar results, mainly that users believed the control interface lacked necessary information the most. It is also shown that more information leads to users believing there was information lacking. These results mostly go against our hypothesis. Although it seems more text makes users more confident which was hypothesized, smaller images contributed more than larger images.

Our third hypothesis that larger pictures would be better for displaying information efficiently and lead to a greater intention of re-use corresponds to questions 1 and 4 respectively. These questions were asked in a positive light so a taller bar is more favorable. For question 1, we can see that a large picture with a short text is deemed as displaying information most efficiently while a large picture and short text is least efficient. This does not confirm our hypothesis as these results for the large picture have a lot of variation. For question 4, the larger images have a greater intention of re-use which is in line with our hypothesis.

Question 3 asked whether participants thought the interface included quality results and didn't follow a particular hypothesis. From the graph we can see that the small image short text seemed to return the best results over all which is interesting because that interface had the least intention of re-use and a low value for ability to display information efficiently.

Figure 4 shows the responses from the participants for the factors they were looking for most while completing the task. Some of these items are as expected such as price and size, but others are not. It would be an interesting experiment to see whether decreasing the amount of text or image to include some of these highly sought after factors on the product page would have a positive or negative overall benefit to the user.

7 Conclusion

This paper studies the effect that image size and amount of text have on users when viewing product recommendations. This is done by outlining the characteristics of a good interface, as well as the best methods used to identify these through a user study. A balanced user study is then designed using these principles and an interface is created that closely mimics what the users would see when shopping online. A user study is then performed with mixed results. Although these results are not statistically significant because of the small sample size, this study takes a strong step forward in identifying the benefit of tailoring a display to mobile devices and concretely outlines a protocol to achieve this.

8 Future Work

The following are the future work that can strengthen the results of the study:

- More tasks should be included, such as purchasing in a category of drones, etc.
- More web scraping would have to be done.
- A larger unbiased participant pool should be used. Several related works have cited the use of Amazon's Mechanical Turk [4]. Enough participants should be used to achieve statistical significance as estimated in Section 6.

In addition, the following were observed:

- Users often focus on the price, reviews, and specific features of the product. Thus, adapting the display of such items could be the focus of future work.
- Some users were unable to tell the difference between some of the interfaces while completing the task.
- All participants completed the task before the time constraint, it may be beneficial to re-evaluate the time allotted to each interface.

9 Division of Work

A **RACI** responsibility matrix is adopted to indicate duties in the project as shown in Table 7. Entities involved may be:

- **R**esponsible for planning and implementing the task.
- **A**ccountable for confirming the completion of the task.
- **C**onsulted for the task.
- **I**nformed of the task progress.

Each member is considered to have contributed equally to the project.

References

[1] Raluca Budiu. 2018. Between-subjects vs. within-subjects study design. https://www.nngroup.com/articles/between-within-subjects/

[2] Linda T Carr. 1994. The strengths and weaknesses of quantitative and qualitative research: what method for nursing? *Journal of advanced nursing* 20, 4 (1994), 716–721.

[3] Fred Churchville. 2021. What is User Interface (UI)? https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI

[4] Vicente Dominguez, Pablo Messina, Ivania Donoso-Guzmán, and Denis Parra. 2019. The Effect of Explanations and Algorithmic Accuracy on Visual Recommender Systems of Artistic Images. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) *(IUI '19)*. Association for Computing Machinery, New York, NY, USA, 408–416. https://doi.org/10.1145/3301275.3302274

[5] Omar Gelo, Diana Braakmann, and Gerhard Benetka. 2008. Quantitative and qualitative research: Beyond the debate. *Integrative psychological and behavioral science* 42, 3 (2008), 266–290.

[6] Justin San Juan. 2022. Recommender System Analysis. https://github.com/justinsj/recommender-system-analysis

Task	Justin	Owen
1. Research		
Literature Review	RACI	RACI
Hypothesis	RACI	RACI
2. Design		
Experiment	RA	CI
Survey	CI	RA
Analysis	RACI	RACI
3. Implementation		
Experiment	RA	CI
Survey	CI	RA
Analysis	RACI	RACI
4. Testing and Evaluation		
Unit & Integration testing	RA	CI
User testing	RACI	RACI
5. Reports		
Proposal Report	RA	CI
Final Report	RACI	RACI

Table 7. Work Distribution

[7] Justin San Juan. 2022. Recommender System Scraper. https://github.com/justinsj/recommender-system-scraper

[8] Justin San Juan. 2022. Recommender System UI. https://github.com/justinsj/recommender-system-ui

[9] Mika Käki and Anne Aula. 2008. Controlling the complexity in comparing search user interfaces via user studies. *Information Processing & Management* 44, 1 (2008), 82–91. https://doi.org/10.1016/j.ipm.2007.02.006 Evaluation of Interactive Information Retrieval Systems.

[10] Suzanne LaBarre. 2018. The design theory behind Amazon's $5.6 billion success. https://www.fastcompany.com/90160960/the-design-theory-behind-amazons-5-6-billion-success

[11] Zameena Mejia. 2018. The surprising reason Jeff Bezos loves bad reviews from 'divinely discontent' Amazon customers. https://www.cnbc.com/2018/04/19/why-jeff-bezos-loves-bad-reviews-from-discontent-amazon-customers.html

[12] Shahin Rahbariasl and Mark D. Smucker. 2019. Time-Limits and Summaries for Faster Relevance Assessing. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 901–904. https://doi.org/10.1145/3331184.3331270

[13] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. 2011. *Recommender systems handbook*. Springer, New York; London.

[14] S.R. Subramanya and B.K. Yi. 2006. User interfaces for mobile content. *Computer* 39, 4 (2006), 85–87. https://doi.org/10.1109/MC.2006.144