Kanoe Design Critique

Team: The PowerPuff Girls

Madison Braman (mabrama@g.clemson.edu)

Date: April 8, 2020

Description and Executive Summary

Description:

The purpose of this document is to evaluate the Kanoe application's wireframe to find usability issues. I evaluated the wireframes using Neilsen's heuristics and then filled out UAR templates which I was provided by the professor. I specifically did nine of these UARs, seven of them being usability problems and two of them being good aspects. These UARs will be given to the team that made Kanoe's wireframes so that they can fix any of the issues they deem reasonable. The UARs are wirtten in sucha way that anybody, familiar with Kanoe or not, should be able to read them and understand them. Following these are a conclusion where I summrize the problems I found and suggestions I have for improving or fixing those issues.

Executive Summary:

My findings were that the team that worked on Kanoe put time and effort into their design. They did a great job overall, but nobody is perfect and can catch everything. Hopefully, the UARs I wrote up will help them improve their application design. I also benefitted from this assignment because I got more experience with looking for flaws in designs and that will help me be more critical when looking at designs I create myself or with my team. Since the team did a great job from the start, I had to be critical and look closely at details to find issues to report back to them. I was able to come up with seven problems and two good aspects to praise, then later on came across another problem I had not seen before.

Introduction

The general purpose of Kanoe as an application is a discount travel website. Kanoe allows the user to search for rental cars, lodging, flights, or even bundles. Each of these has their own section on the website. The user can choose a beginning and end date for each. For example, with the rental cars the user can select when they will be picking up and return the car and for the lodging the user can select when they are checking in and checking out. On each specific search page, there are advanced search options the user can use to narrow down their search results to find what they want more quickly. Once the user finds a listing they like, they can select it to view more detail and possibly make more selections. For example, with a hotel listing, the user may be able to choose between a room with a king bed or a queen bed. For a flight listing, the user can select between economic or business class. After selecting all of the listings the user is interested in, they can get a summary of what is in their cart and fill out their information to purchase the rental car, lodging, and/or flight they had added to their cart.

UAR’s

No. **Madison Braman - 1**	**Problem/Good Aspect** Problem

Name:
No ability to create an account / save user information

Evidence:

Interface aspect: Homescreen and Confirmation/Payment Page

Heuristic: Flexibility and efficiency of use

Explanation:
This heuristic is violated because most travel websites have the ability for users to create accounts, so the user will probably assume that they can make an account. However, there is no place to sign up or create an account which would typically be in the upper bar of a website or suggested again at check out if a user is not logged in. If a user would want to use this website a lot, the user would probably get frustrated having to re-enter their information every time at check out.

Severity or Benefit:

Rating: 3
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):

Frequency: This is a rare problem because most applications these days allow you to create an account and save your user data. All types of users, from new to experienced, will experience this problem because even if the user is a frequent user of the application, they will never be able to make an account to save their data.

Impact: With the current interface, it is low impact. It does not prevent the suer from achieving their goals or properly using the application. At most, the inability to make an account is an aggravation for the user.

Persistence: The persistence will depend on the user. If the user only uses the application on rare occasions,it would be a low persistence. However, if the user wants to use this application numerous times a month, it would be highly persistent because the user would have to re-enter all of their information every time they use the application.

How I weighted the factors: This problem received a rating of three because it is frequent, persistent, but with a low impact. Looking at it from a business perspective, the company could lose many users because the more often a user wants to use the application, the more often they have to enter their information, and the more likely they are to get annoyed with having to repeat that process every time. If the user gets annoyed enough, they will likely start using a different application even if the deals are not as good.

No. **Madison Braman - 2**	**Problem/Good Aspect** Problem

Name:
Switches between “Add to cart” and “Add to Trip!”

Evidence:

Interface aspect: Broad, Bundles, and Flights Search result pages and then the Details page

Heuristic: Consistency and Standards

Explanation:
The heuristic is violated because a button that is supposed to do the same thing on each page is located on is labelled differently on the Details page than the other pages. While the user may not notice, if the user does notice it may create confusion. The user may not know if there is a difference between adding to the cart versus adding to the trip.

Severity or Benefit:

Rating: 1
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):

Frequency: This problem only occurs once in the system with the Details page being different than the rest of them. Users may or may not consider it a problem or notice it at all.

Impact: The problem is low impact because at most it creates some confusion for the user. If the user clicks the “Add to Trip!” button and realized it does, in fact, add it to the cart for them, the problem is solved.

Persistence: The problem is only persistent if the user uses this application numerous times and is bothered by the change. Your average user will likely not be bothered by the change in wording.

*How I weighted the factors:*The problem is only a cosmetic issue because it is simply a change in wording. It is infrequent, low impact, and not persistent. At most, it can cause some confusion at first, but if the user even so much as tries to use “Add to Trip!” in lieu of “Add to cart”, they will realize it works the same.

No. **Madison Braman - 3**	**Problem/Good Aspect** Problem

Name:
Missing cart

Evidence:

Interface aspect: Rental Car, Lodging, Broad, Bundles, and Flights Search Pages versus the rest of the interface

Heuristic: Consistency and Standards

Explanation:
So there is more than one issue with the cart in the upper right hand corner of the above stated search pages. It is never demonstrated or explained what happens when the cart is clicked, but I assume it functions the same way the cart would on other applications: Clicking it would display a preview of what is in the user's cart and could bring the user to their cart if they want. However, the cart is not on all pages of the interface, so the user cannot access their cart from all parts of the application.

Severity or Benefit:

Rating: 3
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):

Frequency: This problem could be considered common. It will mostly impact users who are using the application for the first time because once the user realizes that the cart is only on the above specified search pages, they will know that if the go to one of those pages they can access the cart.If the user is not familiar with the application, they may have to spend some time to figure out how to actually access their cart.

Impact: I would say the problem has a relatively high impact considering the first time a user attempts to use the application they will most likely have a hard time accessing their carts until they realize what pages grant them access. A user would probably find it annoying to have to be on a certain page to access their cart especially if they want to view it while remaining on the page they are currently on.

Persistence: This is a persistent problem that users will continually encounter every time they use the application for the whole duration of using the application. This will become a growing annoyance for the user the more the user cannot access the cart while on certain pages of the application.

How I weighted the factors: Considering this problem is frequent, of high impact, and persistent it is a major problem. The user could waste a bunch of time searching for their cart and possibly not find it or get frustrated and use a different application instead. It is also extremely inefficient to only be able to access the cart from certain pages of the interface.

No. **Madison Braman - 4**	**Problem/Good Aspect** Problem

Name:
Inability to remove things from cart

Evidence:

Interface aspect: Cart/Checkout Page

Heuristic: User Control and Freedom

Explanation:
This heuristic is violated because it is common for users to add things to their cart, find a better deal, add that to their cart, and remove the previously added version. With the inability to remove anything, the user is unable to change their selection.

Severity or Benefit:

Rating: 4
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):

Frequency: This is a common problem that will affect all users whether they are new or experienced because if the user does not know they cannot remove items from their cart and try to remove it later, they will likely get frustrated and not complete the transaction.

Impact: This problem is high impact considering if the user adds one wrong thing to their cart, they most likely will not finish the transaction since they cannot remove their unwanted cart items. It will make the user waste time trying to figure out what to do, make them frustrated, and they will likely give up and not use the site again.

Persistence: This is a persistent problem in the sense that a user may forget they cannot remove something from their cart and add things they do not necessarily plan on keeping. If the user remembers though, they can try to prevent adding anything to their cart they do not plan on keeping.

How I weighted the factors: I would go so far as to say this is a usability catastrophe because it could create a huge problem for the user. If the user ever puts anything in their cart, on purpose or by accident, they can never remove it. This will happen mostly to new users very frequently, it has a high impact, and is persistent. The user will not be able to complete their transaction or will have to waste a lot of money to buy something they did not want. If this is not fixed it is likely the application will not be used once enough users complain about the problem.

No. **Madison Braman - 5** **Problem/Good Aspect** Problem

Name:
Redundant “Popular Locations” on numerous pages where it could be left off

Evidence:

Interface aspect: Rental Car, Lodging, Bundle, and Fight Search pages

Heuristic: Aesthetics and Minimalist Design

Explanation:
This heuristic is violated because on the specified search pages, the user has a specific search goal, so recommendations are just crowding the screen. It is understandable to have recommendations on the home screen though.

Severity or Benefit:

Rating: 1
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):

Frequency: It is not a physical problem that users can interact with, so it oes not really have a frequency. It may bother a few users at most.

Impact: This problem is low impact and easy to overcome. The “Popular Destinations” mini page even has a way to close it if the user does not want to view it.

Persistence: The problem is not persistent because the user can get around it easily or just ignore it if they want.

How I weighted the factors: This problem is only cosmetic. It does not actually cause the user any issues, it can just be distracting or annoying for the user to look at. There is even a way to close it if the user really is bothered by the “Popular Destinations” and does not want to see it. It simply does not need to be on as many pages of the interface as it is.

No. **Madison Braman - 6** **Problem/Good Aspect** Problem

Name:
Selecting dates is difficult due to the method provided

Evidence:

Interface aspect: Calendar Page

Heuristic: Flexibility and Efficiency of use

Explanation:
The heuristic is violated because the user will probably assume they can type in the date they want and not have to choose from a calendar. And in the case of the calendar, there should be a possible drop down menu for the months and years for people who are booking trips awhile in advance. The user would probably get frustrated having to click to go forward a couple of months.

Severity or Benefit:

Rating: 2
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):

Frequency: This problem is experienced by users ranging from new to experienced. It is a frequent problem since the user will have to go through this process every time they use the application for both the leaving and returning date fields.

Impact: This is a low impact problem because it does not stop the user from accomplishing their goal, it just makes the process a little more annoying than it has to be. It may waste a little bit of time, but not enough to consider it a big issue.

Persistence: The user will be bothered by it repeatedly because they have to experience it every time they use the application or want to change the date fields, so it is persistent.

How I weighted the factors: This problem is a minor problem because it is frequent and persistent, but has a very low impact. The user may have to deal with it every time they use the application, but it is not a big enough problem to actually be fixed if there are greater issues at hand. If it were left alone, the application would still work fine, but filling out dates may annoy the user slightly.

No. **Madison Braman - 7**	**Problem/Good Aspect** Problem

Name:
Can have a price range, but cannot sort by prices (high to low / low to high)

Evidence:

Interface aspect: Rental Car, Lodging, Bundle, and Fight Search pages

Heuristic: Consistency and Standards

Explanation:
This heuristic is violated because users commonly want to search for the best deal and therefore would want to sort their search results from the cheapest to the most expensive. While the interface offers a price range filter, there is no way to actually sort from low to high or high to low.

Severity or Benefit:

Rating: 2
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):

Frequency: This is likely a rare or less common problem that users will experience. It will most likely affect new users the most because users who have experience with the application will know there is no way to sort from low to high or high to low. Once a user knows that ability is not present, they will probably find a way to work around it.

Impact: This problem has a pretty low impact because at most, user's will just waste a bit more time going through the search results to make sure they find the best deal for what they are looking for. The lack of the sorting option does not hinder the user from achieving their goal at all.

Persistence: While the inability to sort the prices may bother a user, they can always work around it. It is a persistent problem though considering there is no other way to fix it than for the user to mess around with the price range they search with.

How I weighted the factors: Overall, the problem is not an urgent one and could continue to be left off. Adding the functionality of being able to sort the prices from low to high or high to low would greatly improve the user's experience because it is probably very common for user's to sort prices on an application such as this one. Considering it has a lower frequency, a low impact, but is persistent, this problem would be a good one to fix.

No. **Madison Braman - 8**	**Problem/Good Aspect** Good Aspect

Name:
The wide range of advanced search filters

Evidence:

Interface aspect: All Search Pages and Search Result Pages

Heuristic: User Control and Freedom

Explanation:
This heuristic is adhered to because users may know exactly what they are looking for and being able to shorten search results by getting rid of listings the user does not want to see makes their search a lot more efficient.

Severity or Benefit:

Rating: NA
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):
Frequency:
Impact:
Persistence:
How I weighted the factors: It is very common for the user to have specific wants in mind, so being able to specify those wants and hide results they do not care about makes the user experience much more pleasant. It also helps the user find what they are looking for much more easily than having to search through all the results they could have without any filters on.

No. **Madison Braman - 9**

Problem/Good Aspect Good Aspect

Name:
The ability to search for specifically rental cars, lodging, flights, or bundle deals

Evidence:

Interface aspect: All Pages / the entire interface

Heuristic: Flexibility and Ease of use

Explanation:
This heuristic is adhered to because having the option to search for a range of things on one application is extremely user friendly. It is especially useful because users will probably have to book more than just a rental car or just lodging while traveling, so being able to book them all on the same application rather than numerous different ones is more manageable.

Severity or Benefit:

Rating: NA
NA – good aspect
1 – cosmetic problem (does not matter too much)
2 – minor problem (would be nice to solve, but not a high priority)
3 – major problem (a problem that should definitely be solved with high priority)
4 – usability catastrophe (a problem that renders the Web site / app useless)

Justification (Frequency, Impact, Persistence):
Frequency:
Impact:
Persistence:
How I weighted the factors: When a user is traveling, they will typically need to book more than just a flight, lodging, or a rental car, but likely a combination of the few. Being able to book and buy them all on the same application is more convenient for the user. It is even better because the application is split into sections depending on what the user wants to search for.

Final Conclusion

About half of the problems I reported were in the search pages or the search result pages. These include: Having a price range, but no way to sort by price, the “Popular Locations” sidebar crowding the pages it does not need to be on, the missing cart in the upper right corner, and inconsistent button labels. The biggest problem out of that list is the missing cart rendering users unable to get to their cart from certain pages. The other problems I found include: the inability to create an account or sign up for one, the inability to remove things from the user’s shopping cart, and the difficulty of choosing dates on the calendar page. I would say that these are the bigger problems that should be fixed first, especially the inability to have an account, the missing shopping cart, and the inability to remove things from the user’s cart. These three things specifically make the application harder for users to use while the rest are more on the side of minor or cosmetic issues.

My recommendations for improving this application’s design would be to remove the “Popular Locations” on the pages I specified in UAR “Madison Braman – 5”, add the shopping cart in the upper right hand corner of the application to every page that is missing is (reference UAR “Madison braman - 3”), and add the functionality of removing deals or listings from the user’s cart. While it is not necessary for users to be able to create an account, giving them the option to would greatly improve Kanoe. Any other cosmetic issues I mentioned in my UARs would be worth fixing and I had specified what needs to be done in the UARs. Besides a few things that should be fixed, the team that built Kanoe did a great job and I am sure they will improve their design after getting this feedback.

Reflection

How many problems did you find?

It was difficult to find problems because the overall design is good, but as I was writing the report I realized they do not have a back button. I do not have time to add a UAR in for that, but it is something they should probably add in. So, in total I found about eight issues that could be fixed.

Do you think you found all problems, or were there more beyond the ones you reported?

As stated above, the lack of a back button is an error I did not report because I did not notice it until later. Keeping that in mind, I am sure there are probably more errors I did not catch because I am only one pair of eyes, so I probably missed a few things.

How much time did you spend on the evaluation and finalizing UARs?

I spent approximately six hours evaluating and finalizing the UARs because I struggled to find issues the first look through. I had to really evaluate and look at their wireframes to find issues because there were no major problems that jumped out.

Do you think Neilsen's Heuristics are a good tool for finding usability problems? Why (not)?

I do think Neilsen's heuristics are a good tool for finding usability problems because it covers almost all the problems a user could run into when using an application. They also provide a guideline of what kinds of issues to look for when evaluating something like a wireframe.

Do you think that UARs are a good way to describe usability problems? Is there anything you would add to the UAR template to make it better?

I personally think that the UAR template works well for evaluating usability problems because it covers what the problem is, where it is, explains why it is a problem, and then goes further into detail about how severe of a problem it is. I cannot think of anything that I would add to the UAR template because this was my first experience with it and it worked well for this evaluation.

How would you improve the process if you had to do the same assignment again?

Considering I feel like I was unable to find all the issues on my own, I think this would have been better as a group assignment. I know due to COVID19 that it is rather difficult right now to do group work and my teammates may have found the problems I did not, I feel like being able to talk out loud with others and walk through the wireframes together would have made the process easier and helped me find more errors.