Interaction analysis 4/7

Continuing with my previous project; a b2b HVAC-wholesale. In my second essay I defined that that the system is the webstore (and the management system within) and the interface used in a physical warehouse. With this project we are lucky. We get to do continuous development on this system. So, we get the user feedback all thought it is "filtered" by our contacts in the company who oversee the web development.

In the second essay I concluded that for example the placement of the login button could be considered a scripting of the interactor. As in the designer might place the login button in the spot, they think the user will most likely look for it. Or they might choose the spot based on their own previous experiences.

The most easily observable cultural influence in the system I'm analyzing is the flow from left to right and top to bottom. Regarding the case of the login button, it is placed in the upper right corner in this system webstore side. The thought of things flowing from left to right is so prominent in western culture that visual art with people facing right is more easily read as "hopeful" and "looking to the future" even by observers who are not familiar with basic evaluation of visual art. In this system all the buttons leading to a next phase, page or step are on the right and button leading back are on the left. This flow in obviously no universal and might be a hindrance for big enterprises that extend their operation worldwide. Big enterprises often have "in house" development teams which might not be very multicultural.

Names are strongly tied to language and culture. A 2018 article by Tony Rogers addresses beliefs programmers might have about names and how these beliefs are wrong or have exceptions.

(T. Rogers, 2018, Falsehoods Programmers Believe About Names -With Examples, https://shinesolutions.com/2018/01/08/falsehoods-programmers-believe-about-names-with-examples/).

How names work might seem very self-evident and obvious to people and it might even be hard to pinpoint these deficiencies in ones thought just with introspection. Introspection could even be a hinderance in this situation as programmers like to make controls and inspection in their code to make it more secure. A programmer might congratulate them self on a good job when they made a check that doesn't let you send a last name-field with more than 2 names in it. What the programmer doesn't realize is that they are making a bad user experience for Mr. de la Cruz. Or another programmer might try to eliminate spam and unwanted requests by making the first name field check for "bad words" and end up censoring someone's name. The ideal model would be the system allows all types of inputs to register at all situations, but the represented model is far from this.

In this HVAC-wholesale system there was an information flow problem. It was because when one field was wrong in one database another database didn't accept any information regarding that order. In this case if the free word "additional information" box contained more than 255 characters the other database rejected the data. This problem emerged because the creators of the second

database didn’t imagine that field to be used the way it was used or the end user needing more than 255 characters for their additional info. In this case a prejudiced point of view might have made the program run more smoothly if the HVAC system programmer had restricted the character amount right at the beginning the same way as the second databases designers.

In this essay I came up with prejudices that hinder design. In all: all prejudice takes the system away from the ideal model. But users’ possibilities need to restrict and checked to make the system work in intended ways. Email needs to be a real email and a name must be given to verify credit. In my work prejudice about the user’s ability understand what is important and what is not is always present. This view could be seen as result of my own prejudices. I have adopted a ‘if the user can use it wrong, they will’ -attitude from my more experienced colleagues. I restrict and force the user to behave in certain way because it would be inefficient than to motivate the user to act the ‘correct’ way in other ways. One emergent behavior on an inventory functionality I did was that the user called in helpers to do the inventory. They all used one person’s login information and it almost broken the database used for this. After this case my prejudice about the user has made me think more what could go wrong and prepare more for this.

Self-Evaluation

My motivation wasn't as high for this essay as it was for the previous ones. and I feel it shows in the result. I didn't spend as mush time on this as I did on the previous ones. I might change my project for the next essay as this one doesn't inspire me as much.

Structure: 2. I have a beginning end and a middle clearly

Clarity: 1.5, My thought here a little all over the place. I have clear point but id like them to tie in together a bit more. I get my thought across thought.

Content: 1.5, I analyze all the parts I talk about clearly even thou they could have been more tied to together.

Evaluation/criticism: 1, In the last paragraph I do a little introspection on my own work in the system and how prejudice might prevent problem situations. Some tips can be read from the text but are not explicitly spelled out as improvement. As an example, for one of these tips: how relate to people's names.

Big picture: 2, my essay focuses more on the general side as I found it more interesting. I tie in certain small aspects of the system working on. I found it more interesting to talk about how big cultural things can affect small buttons and text fields.