Implementation of Usability Principles

In the development of application design principles have emerged as an agreed upon convention to steer the design process of new apps. A set of principles called the Usability Heuristics by Nielsen have acquired a remarkable importance in the field of Usability principles. Hence, they were a main guideline to develop the HiFi app prototype for the menu app. Each of Nielsen's ten heuristics were applied to the app. Furthermore, our implementation is explained.

#1: Visibility of system status

The principle entails the idea that the internal state of the system should be visible for the user. But what state is relevant to the user? Obviously, the user needs help in the process of *choosing* (Which ingredients does this food contain? Can I eat this? What does this match with?) *ordering* (What did I order? Is food being delivered? How much does this menu cost?), and in the process of *paying* (How much will I have to pay? How can I pay?).

All these states of information should be visible for the user. How were they implemented? This can be well explained by looking at the food *choosing* process. To avoid visual clutter, only the most relevant states are transparent in the menu, such as pricing. More information can be found by looking at the detail view.

#2: Match between system and real world

The developed app should always speak the language of the user, using this heuristic. Accordingly, jargon of dining should be avoided or at least directly linked to an explanation. In our design, this was considered in the explanatory texts in the food view, describing what is hidden behind the name.

Not only does the rule include language, but also familiar concepts: Plenty can be found for restaurants (the restaurant menu, the waiter's pad, the served table) and some of those were used in the paper prototype. We are planning on profiting from the full scope of analogies and thereby improving the familiarity and usability for the user.

#3: User control and freedom

It is very frustrating for users to do something and not being able to return or reverse the action made. That is why we strived to make every screen accessible at any point of the visit, if it makes sense or is still possible. Irreversible states are frustrating especially when they have real-world consequences like not ordering what you desire. Although not displayed in the prototype, orders can be aborted if not already in the process of cooking.

#4: Consistency and standards

Using an app often includes learning new information. To improve our app's learnability, internal and external consistency is to be considered. To ensure internal consistency, selected symbols and button designs are unaltered while using our app.

External consistency, focusing around following conventions, is respected from the beginning of development in this app. We asked our users, how they recall their visit chronologically. We tried sticking to that procedure and only altering, where the user profits from it (Like ordering before going to the restaurant to have more quality time).

#5: Error prevention

When speaking about errors, it is important to understand what errors are meaningfully bad. The first being ordering something wrong. Ordering something wrong can be prevented by transparently showing the users what they added to the ordering list ("the shopping cart"), what we achieved in the overview section. The second worst thing being deleting the order list mistakenly, forcing the user to reenter everything. Here, clear

warnings will have to be implemented in the further design. In the following iterations of evaluation, unpredictably occurring errors will be observed, and countermeasures implemented.

#6: Recognition rather than recall

Instead of forcing users to remember what to do, and which action is hidden behind which, we should rather strive for steady clarity. Therefore, we avoid using unexplained symbols and used buttons including text (e.g. the "ORDER" button).

#7: Flexibility and efficiency of use

Especially for experienced users, there should be ways to reach their goals faster. These accelerators were used in the quick-add button to order something without further ado. This saves time for regular guests while not sacrificing the availability of further information.

Personalizing the app also helps in accommodating the usage: In the check-in screen, users can for instance specify their nutritional habits (vegan, pescatarian, keto, etc.) as well as food allergies. These information can be used to filter out settings, show warnings if food does not comply with the diet, or pre adjust settings (e.g. setting sharpness to zero when user doesn't like it)

#8: Aesthetic and minimalist design

Throughout the app interface, we tried minimizing unnecessary elements and maximizing the amount of useful information. In the following prototypes, we will need to improve especially in this section, as our app might still look a bit overcrowded.

#9: Help users recognize, diagnose, and recover from errors

Recalling the identified "bad" errors from #5 (deleting the order list and ordering something wrong) we can try to build a system that avoids negative consequences from these errors. Deleting can be avoided by warning messages or a button that reloads the last list.

#10: Help and documentation

Help can be divided into proactive and reactive help, according to the heuristic: Proactive and reactive help. Reactive help is aimed at helping the user when stuck. Fortunately, in a restaurant, there is an actual waiter that will be happy to help, when users get stuck. Therefore, we included an "call-a-waiter" button.

Proactive help is there to familiarize people with an interface and is given in the form of push revelations (context-independent tips) and pull revelations (context dependent). Both types are not yet implemented – still we have some concepts in mind: A virtual waiter's recommendation to order matching food and an initial tour through the app.

Sources

https://www.nngroup.com/articles/ten-usability-heuristics/

https://www.nngroup.com/articles/help-and-documentation/