

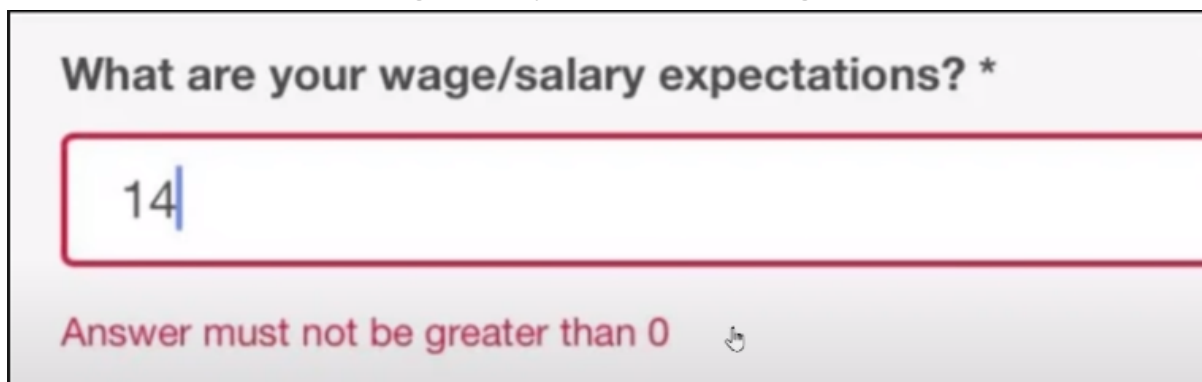
Złe interfejsy wokół nas

Dobry interfejs przedmiotu lub programu pomaga użytkownikowi w osiągnięciu celu, podpowiada kolejne kroki, rozprasza wątpliwości. Zły - wprowadza konsternację, rozprasza i uniemożliwia wykonanie czynności. Projektując interfejsy należy mieć te zasady na uwadze, ale mimo to, czasem się nie udaje.

Przyczyn jest wiele: chęć zrobienia bezpiecznego urządzenia lub programu, chęć bycia oryginalnym, optymalizacja rozmiaru urządzenia, albo też brak wiedzy jak można zrealizować oczekiwaną funkcję w inny sposób.

Na forum (<https://www.reddit.com/r/BadDesigns/>) poświęconego złym projektom znalazłem kilka przykładów dobrze oddających powyższe przyczyny:

1. Chęć zrobienia bezpiecznego urządzenia lub programu



What are your wage/salary expectations? *

14

Answer must not be greater than 0

Rysunek 1. Weryfikacja wprowadzanych danych. [dostęp: 20.10.22]. Dostępny w Internecie:

https://www.reddit.com/r/BadDesigns/comments/kl1slm/of_course_im_gonna_work_for_free/

Projektując obecnie programy lub strony internetowe przeznaczone do wprowadzenia danych przez użytkownika należy bardzo uważać na ataki hakerskie. Używając pól tekstowych można zastosować atak "SQL injection", aby się przed nim zabezpieczyć, zawartość pola tekstowego jest sprawdzana czy nie zawiera znaków specjalnych (np. średnik), albo czy zawierają odpowiedni typ danych (np. w polu 'Wiek' mogą być tylko liczby). Czasami, jak widać na zamieszczonym obrazku, zabezpieczenia są źle dobrane, co prowadzi do problemów.

Rozwiązanie: poprawienie weryfikacji danych, aby pole przyjmowało tylko liczbę naturalną.

2. Chęć bycia oryginalnym



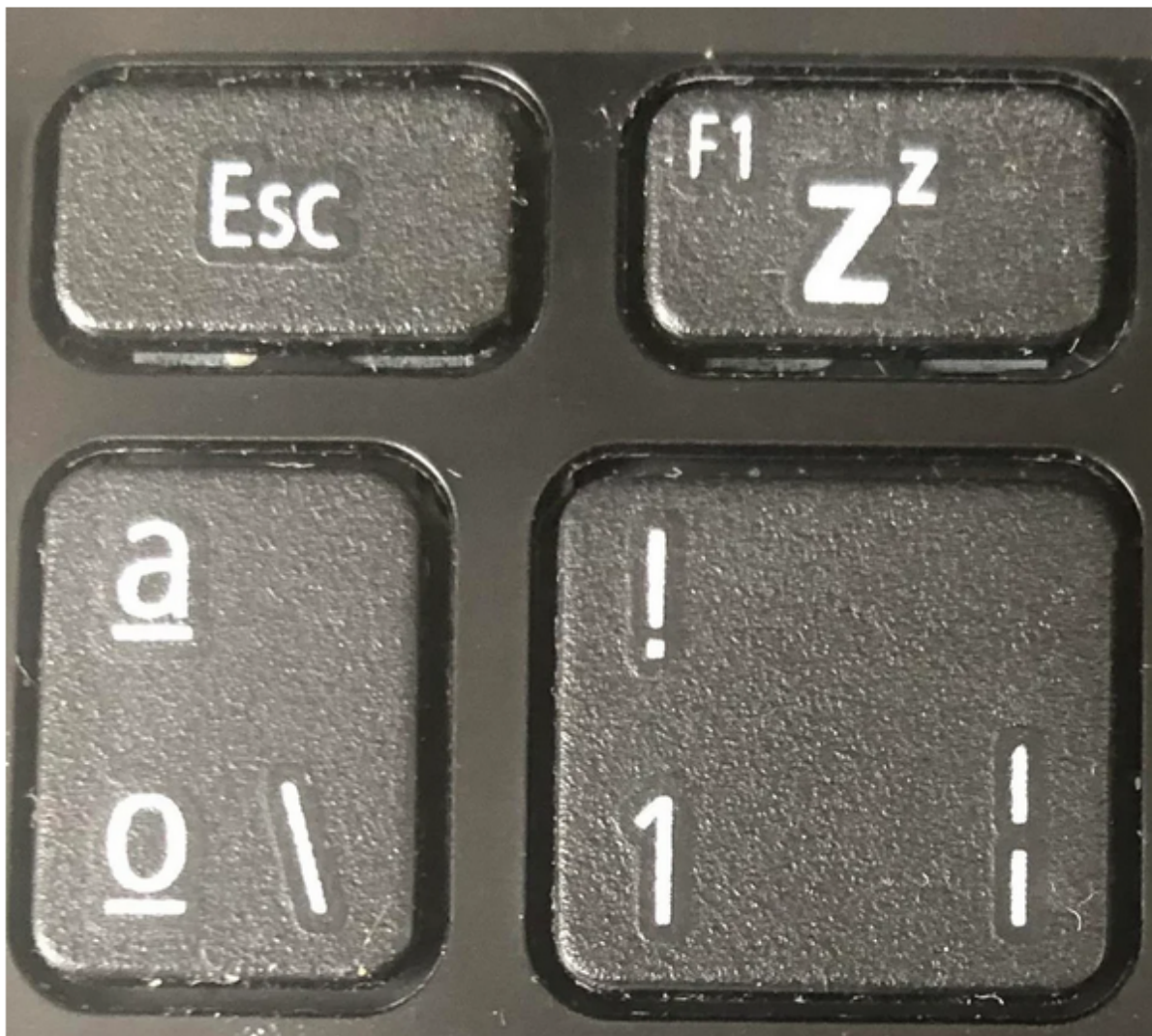
Rysunek 2. Kubek. [dostęp: 20.10.22]. Dostępny w Internecie:
<https://www.reddit.com/r/BadDesigns/comments/njh5tr/humansheadache/>

Jest to przykład projektowania przedmiotu, aby pełnił wyłącznie funkcję dekoracyjną - ignorując potrzeby użytkownika. Niestety, tego typu produkty często są bezużyteczne, ponieważ nie zostały zaprojektowane z myślą o ich używaniu.

Rozwiązanie: zmiana kształtu albo dodanie warstwy przezroczystej żywicy, która pozwoliłaby ujednolicić kształt kubka, pozostawiając widoczny kształt.

3. Optymalizacja rozmiaru urządzenia

I wanted to press escape and accidentally pressed sleep at least 20 times



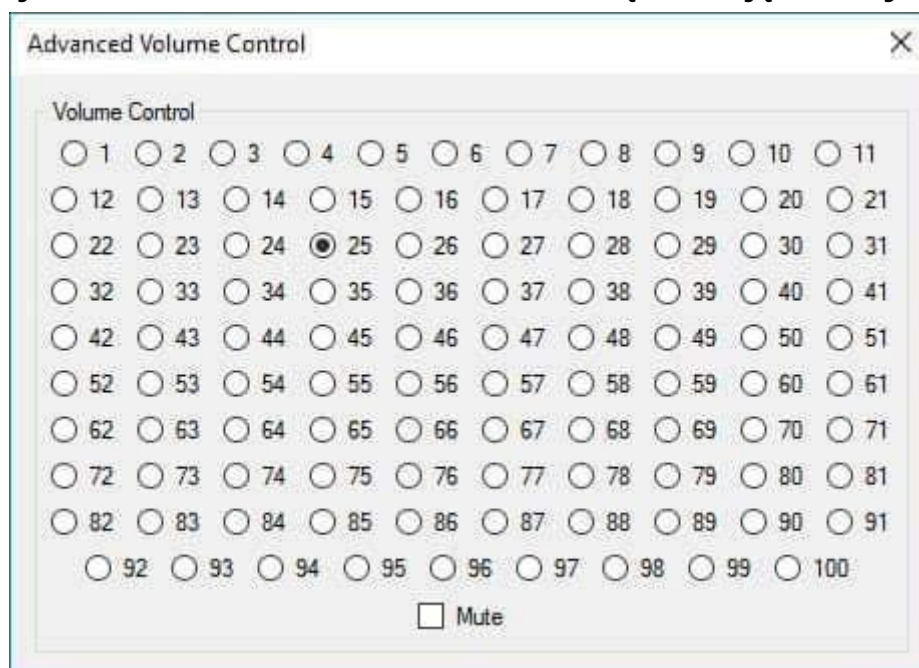
Rysunek 3. Przycisk usypiania komputera. [dostęp: 20.10.22]. Dostępny w Internecie: https://www.reddit.com/r/BadDesigns/comments/owbsgo/i_wanted_to_press_escape_and_accidentally_pressed/

Jednym z trendów rynkowych jest miniaturyzacja sprzętu komputerowego. Chcąc tworzyć coraz lżejsze i mniejsze laptopy, należy też zmniejszać klawiatury. W tym celu, niektóre przyciski muszą spełniać więcej niż jedną funkcję.

Pomijając fakt, że takie wielofunkcyjne przyciski są często nieintuicyjne i stanowią nietrafiony interfejs same w sobie, w powyższym przypadku mamy do czynienia z zupełnie innym problemem. Obok klawisza ESC, który jest dosyć często używany w różnych aplikacjach, projektant umieścił klawisz usypiania komputera. Dodatkowo, cały górny rząd klawiszy jest mniejszy od pozostałych. Użytkownicy są przez to narażeni na przypadkowe naciskanie niewłaściwych klawiszy.

Rozwiązanie: projektując klawiaturę nie należy ustawiać często używanych funkcji obok wyłącznika urządzenia lub podobnych funkcji. Należy wydzielić przestrzeń specjalnie dla tych klawiszy z dala od często używanych funkcji.

4. Brak wiedzy jak można zrealizować oczekiwaną funkcję w inny sposób



Rysunek 4. Sterowanie głośnością. [dostęp: 20.10.22]. Dostępny w Internecie: https://www.reddit.com/r/BadDesigns/comments/ma502o/ui_design_is_my_passion/

Od wielu lat zmiana głośności urządzenia jest przedstawiana przy pomocy suwaka, jest to dobrze przetestowany interfejs, który stał się tak naprawdę standardem. Możliwe są tylko dwa powody, aby nie korzystać z takiego utartego interfejsu:

1. Wierzymy, że nasz interfejs jest lepszy i zastąpi ten używany do tej pory.
2. Nie wiemy jak w danym języku programowania czy danym środowisku wdrożyć takie rozwiązanie, więc radzimy sobie jak tylko potrafimy.

Rozwiązanie: poświęcenie więcej czasu na dokładne zapoznanie się z językiem programowania i środowiskiem, aby wykorzystać dostępne kontrolki we właściwy sposób.