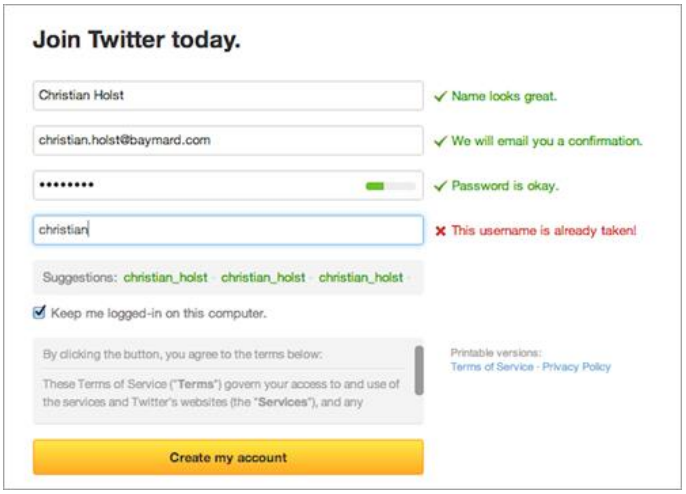
Web Form Usability Guidelines

1. Live inline validation



Validating the input before the user has submitted the forms prevents him from entering his details all over again due to late error notification. Each form field is validated separately as the user types. The error handling is most often instant, with the user being told that their data doesn't match the expected format (although the user can scroll past and try to submit the form anyway).

2. Remove all unnecessary information

Commented [k1]: Changed to lowercase for consistency with other headings.

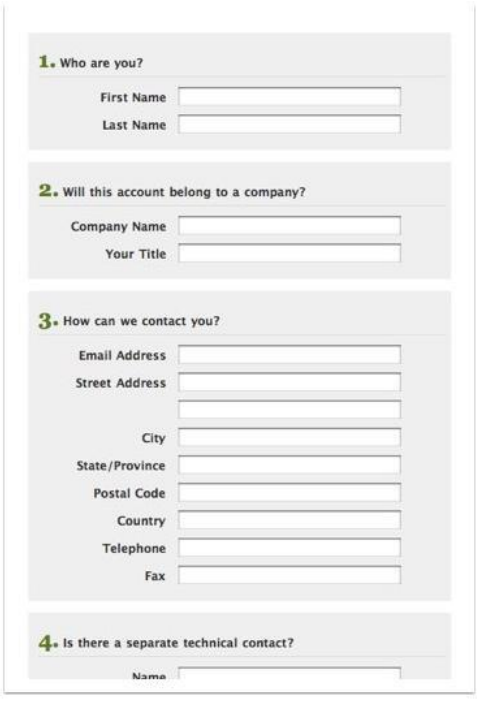
No one likes to fill out long web forms, so excluding any irrelevant information is a must. The more form fields are displayed on the screen the more the user perceives he has a lot of work to do, and this increases the cognitive load, which leads to avoidance of the task and frustration. For example, for the address use only one form instead of two. The smaller the amount of web forms the less time the user will spend completing them. We should also aim at removing any clutter and leave more white space for better readability.

3. Display hints and examples

Users might not understand what they have to input in a particular field or have uncertainties. Providing additional helpful information decreases user ambiguity. If you expect a specific format, make sure to give an example. Giving examples is very useful to users.

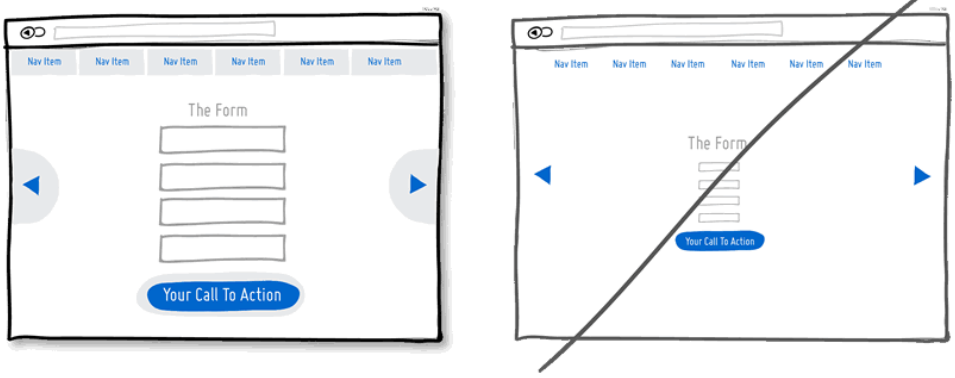
4. Divide web forms into sections

Visually breaking a form into sections will help the users semantically understand the variety of questions we're asking them. For example, an insurance auto quote could be divided into personal information, information about the car, information about the driver, etc.



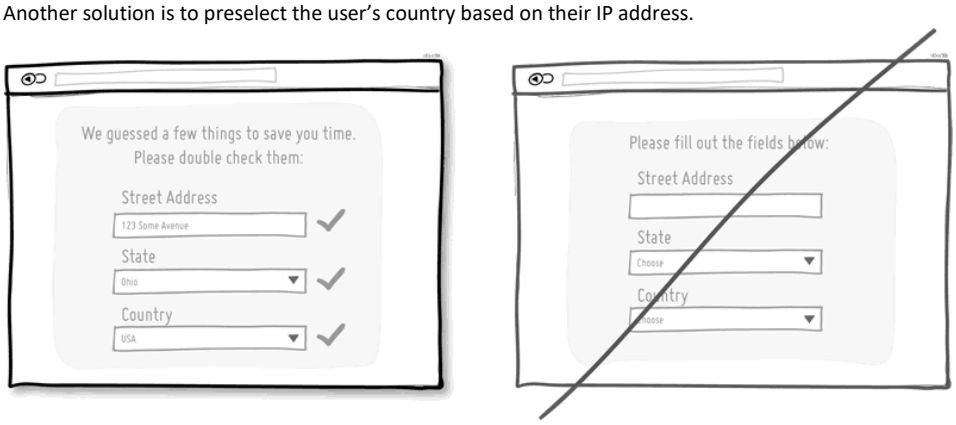
5. Form size and spacing

Use 14px+ font for better readability and provide wider form fields for easier tapping and selecting with a mouse. Forms and buttons can all be made easier to click on if their size is increased. According to Fitt's Law, we need more time to click on something with a pointing device the farther away it is and/or the smaller it is. For this very reason, do consider increasing your form fields and calls to action. Make sure to give your respondents plenty of space, both vertical and horizontal. Adding space between items will make the form more easily readable and much less cognitively overwhelming.



6. Auto-format when possible

Save the user credentials and then autofill after the first submission attempt. Again, if the user cannot complete the forms the first time we should **avoid making him fill in the same forms again**. It causes frustration and takes more time to complete. Also, use default values for some fields. For instance, if most of the customers are from Bulgaria, make the form field "Country" set to default to Bulgaria. Another solution is to preselect the user's country based on their IP address.



7. Make use of HTML5

Use predefined fields for browser side validation

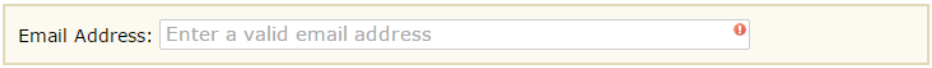
HTML5 added several new input types like date, email, search, number, range, time, and URL. They can be validated by the browser eliminating requests to the server. Some smartphones recognize the email type and add ".com" and "@" to the keyboard to match email input and generally better touch interaction is supported by these fields. The downside is that not all browsers support the new input types; however, if this is the case they will behave as input type "text".

Use autofocus HTML5 attribute on the first form

The autofocus attribute makes the specified field selected and ready for typing when the page is loaded. This way some time is saved in selecting the first field with the mouse or mobile device. Also, there was a study that suggested faster form completion times when this technique is incorporated.

```
First name: <input type="text" name="fname" autofocus>
```

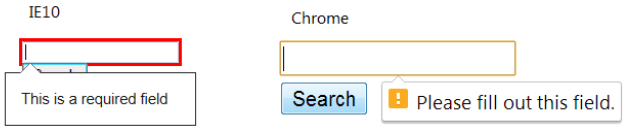
Use a placeholder



The placeholder attribute lets us display a prompt or instructions inside the field (the grey text); something that previously had to be implemented using messy onfocus and onblur JavaScript events.

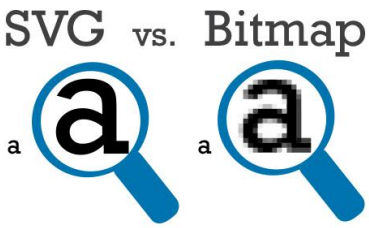
Use the required attribute

Instead of using server side validation, use the required attribute to validate input by the browser.



Use Scalable Vector Graphics (SVG) where possible

SVG are XML based vector graphics that don't lose clarity while scaled and require much less bandwidth than bitmap graphics. They offer perfect quality, even when zoomed in, and decreased page loading speed.

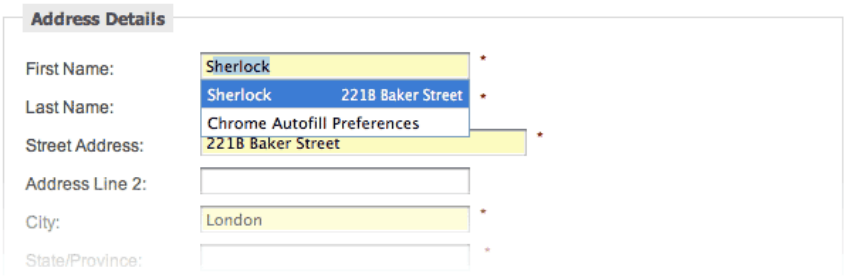


8. Use visualizations to support working memory

20% of the human brain is wired for processing visual cues, therefore visual representation of objects (cars, boats) is processed much faster than text only. For example, when users report a claim, it's much easier to point out on an image of a car which parts exactly were damaged rather than using checkboxes only. Problems with naming conventions are also eliminated this way.

Commented [k2]: Referring to "visual representation", singular.

9. Use common labels like fname, email, and lname

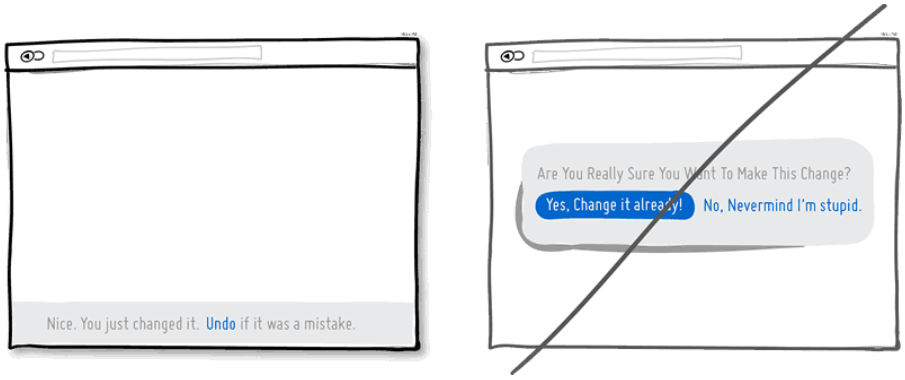


In order to support automatic client-side autofill on forms (Chrome autofill, Firefox plugins, etc.) we should use a common "ID" like fname for a first name. This can reduce significant time when filling in the forms because the user's browser will do the work.

Commented [k3]: The period after etc tells us it's an abbreviation of etcetera.

10. Give users an undo function to roll back changes

Sometimes users might enter erroneous information and want to go back and undo their progress. It is vital to provide an undo option as it is listed in one of Nielsen's 10 usability heuristics.

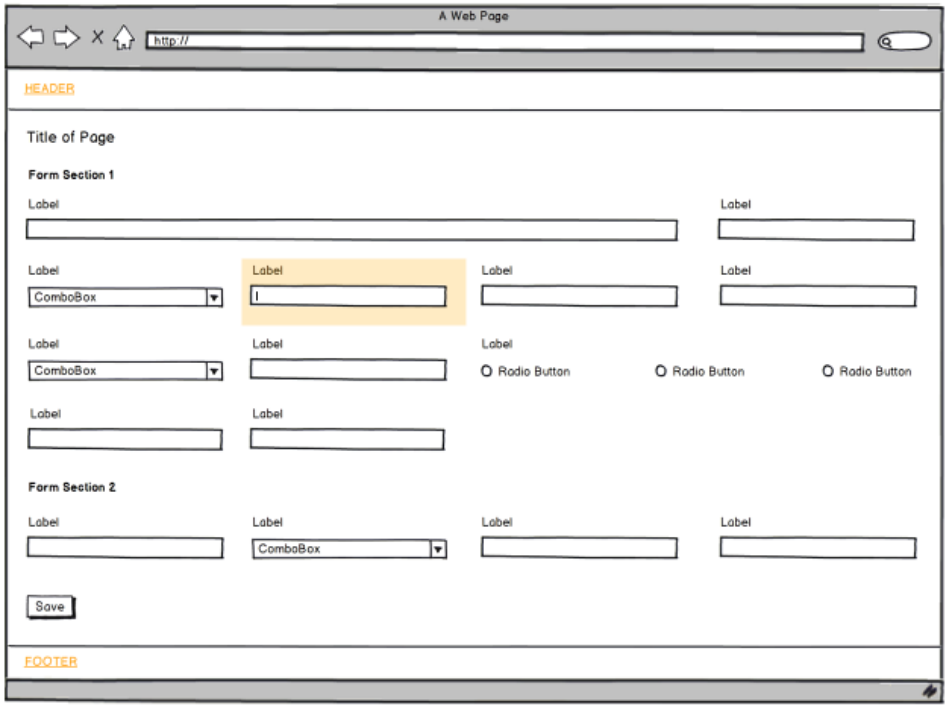


11. Tabulation is very important

Expert users make use of the TAB button to move quickly through form fields. We should make sure that we provide a sequential order of fields so that the users are not confused as to where their tabulation went. This way, expert users will be able to fill out the forms faster than manually selecting each field with the mouse.

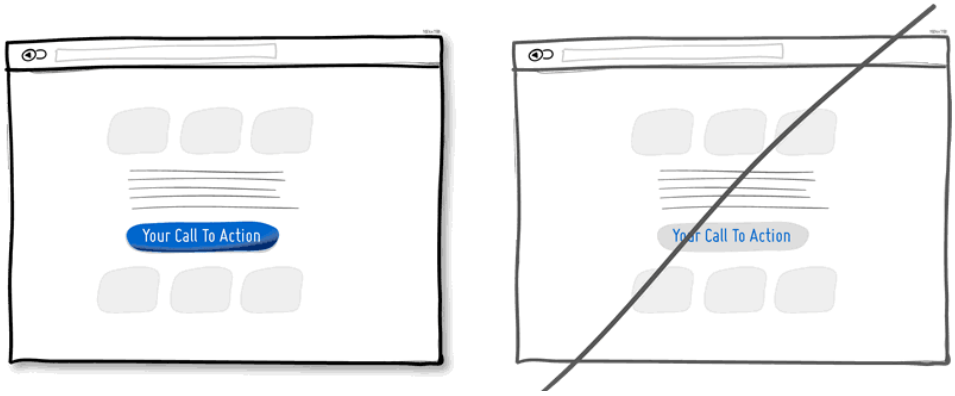
12. Highlighting the active form field

When users are filling out complex web forms they often lose focus on which form field they were filling out and start visually searching for it. An elegant solution would be to highlight the active form. This way, users will return to work faster after an interruption or break.



13. Clear call to action

Make the next action in a task clearly visible and contrasting to the other UI elements to support efficient task flow. For example, when the user can easily spot the “Submit” button after filling out the web forms, they spend less time searching for the button.



14. Thank the user and provide useful information after task completion

After the web form is finished, a thank you page with additional information on when users will receive a response or other important information should be provided. The user has to know what to expect next.