



Experiment No. 6

Title: Patterns with form design



Batch: B-2

Roll No.: 16010422234

Experiment No.: 6

Aim: To create patterns with form design

Resources needed: Wireframing tool

Theory:

The patterns, techniques, and controls described here apply mostly to form design—a form being simply a series of question/answer pairs. However, they will also be useful in other contexts, such as for single controls on web pages or on application toolbars. Input design and form design are core skills for interaction designers. Following are some samples of form design:

Forgiving Format:

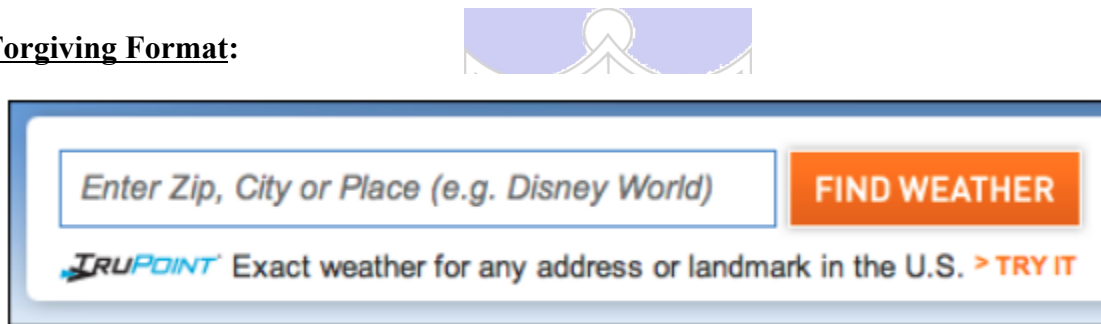


Figure 1 Forgiving Form Format

Forgiving form format permits users to enter text in a variety of formats and syntax, and make the application interpret it intelligently.

Use When:

UI asks for data that users might type with an unpredictable mix of whitespace, hyphens, abbreviations, or capitalizations. More generally, the UI can accept input of various kinds from the user—different meanings, formats, or syntax. Examples:

The New York Times uses Forgiving Format in several features that need information from users. Figure shows examples from its real estate search and from its financial quotes feature.

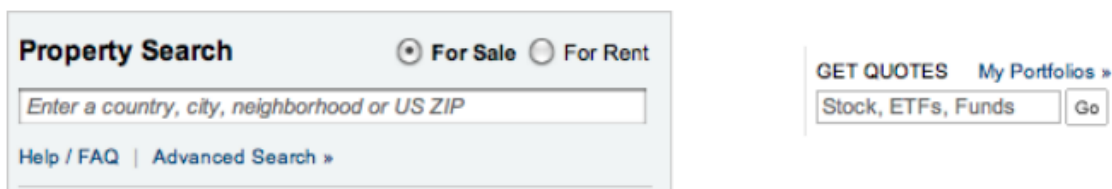


Figure 2: Two text fields in the New York Times website

One place where this pattern should be used, but usually isn't, is when credit card numbers are requested from the user. As long as 16 digits are typed, why should the form care whether the user separates them by spaces, or by hyphens, or by nothing at all? It's not difficult to strip out separating characters. PayPal, for example, doesn't accept spaces in credit card numbers shown in figure 3.

Figure 3 PayPal Form



Figure 4 comes from Outlook's tool for setting up a meeting. Look at the "Start time:" and "End time:" fields at the bottom of the screenshot—one don't need to give it a fully defined date, like what's in the text fields now. If today is April 24 and one wants to set up a meeting for April 29, one can type any of the following terms:

- next Thu • 29/4/2004 • 4/29
- nxt thu • 4/29/2004 • five days
- thu • 29/4 • 5 days

And so on—there are probably other accepted formats, too. The specified date then is "echoed back" to the user in the appropriate format for the user's language and location.

The screenshot shows a standard Outlook event creation window. It includes a 'To' field with a dropdown arrow, 'Subject', 'Location', 'Start time' (with a clock icon), 'End time', and an 'All day event' checkbox. The start and end times are set for Thursday, 4/29/2004, from 8:00 AM to 8:30 AM.

Figure 4 Outlook's tool form

Structured Format:

The screenshot shows a form with three main sections. The first section has a 'Name:*' label followed by a single text input field. The second section has a 'Company:' label followed by a single text input field. The third section has a 'Serial Number:*' label followed by a series of six small, individual text input boxes. The first box contains the number '1045', and the second box is highlighted with a blue border.

Figure 5 Structured Form Format 1

Instead of using one text field, use a set of text fields that reflect the structure of the requested data.

Use When:

Form interface requests a specific kind of text input from the user, formatted in a certain way. That format is familiar and well defined, and designers don't expect any users to need to deviate from the format designer expects. Examples include credit card information, local telephone numbers, and license strings or numbers. It's generally a bad idea to use this pattern for any data in which the preferred format may vary from user to user. Consider especially what might happen if the interface is used in other countries. Names, addresses, postal codes, and telephone numbers all have different standard formats in different places. Consider using Forgiving Format in those cases.

Examples:

At its simplest, Structured Format literally can take the shape of the data, complete with spaces, hyphens, and parentheses, as illustrated in the following figure.

Telephone number	(504) 555-1212	(504) 555 - 1212
Credit card number	1021 1234 5678 0000	1021 1234 5678 0000
Date	12/25/2004	12 / 25 / 2004
ISBN number	0-1950-1919-9	0 - 1950 - 1919 - 9

Figure 6 Structured Form Format 2

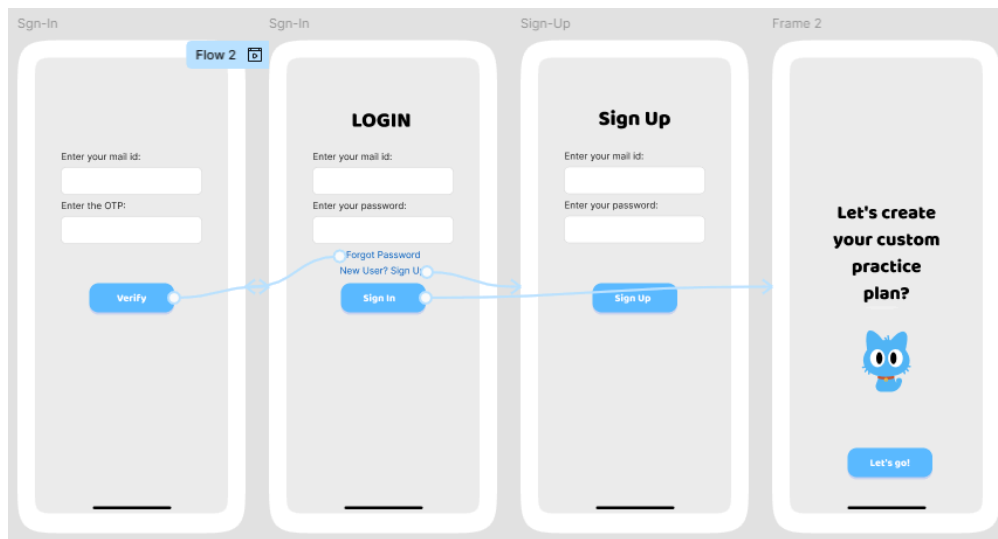
For date input, LiveJournal uses Structured Format in combination with a drop down to choose a month (see Figure 7). It defaults to the current day and time.

Date: , : (24 hour time)

Subject:

Figure 7 Structured Form Format 3**Procedure:**

Create wireframes incorporating patterns with Form design for the chosen topic

Result:

Outcomes: Apply principles of Web interface design

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

In this experiment, we successfully applied forgiving and structured formats in form design through wireframes, enhancing user experience by accommodating diverse input methods and ensuring data accuracy. The forgiving format allowed flexibility for unpredictable user inputs, while the structured format provided clarity for specific data requirements. This exercise reinforced the importance of thoughtful input design in creating intuitive and user-friendly interfaces, ultimately improving overall usability in web applications.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

1. Tidwell, Jenifer, "Designing interfaces: Patterns for effective interaction design. " O'Reilly Media, Inc.", 2010.
 2. Wilbert O. Galitz, "The Essential Guide to User Interface Design - An Introduction to GUI Design Principles and Techniques", Wiley Computer Publishing, Second Edition, 2002
 3. Bill Scott, Theresa Neil, "Designing Web Interfaces Principles & Patterns for Rich Interaction", O'rielly Media, First Edition, 2009
-