

REORGANIZING AND PRESENTING DATA FIELDS WITH ERRONEOUS INPUTS

Junze Yu

BACKGROUND

5 Field

[0001] The present invention relates to the technical field of data processing, and in particular, relates to a method and system for reorganizing and presenting data.

Related Art

10 [0002] For web pages, a common input method for a user to enter data is to use web forms. A web form can include a plurality of data fields. In particular, the widely used Bootstrap web form can accommodate as many as 50 to 100 or even more data-entry fields.

[0003] When entering data, a user could make mistakes. For example, the user might use an invalid input format, exceed the limit of the number of characters, or leave one or more input
15 fields blank. When the user completes the form data entry and submits or stores the form, the system typically verifies the data entered by the user. If an input error for any data field is detected, the system typically does not allow the form to be submitted or stored. In this case, the system may return to the form page where the original data fields are presented and request the user to re-enter the data fields for which errors have been detected (i.e., have erroneous inputs).

20 [0004] Typically, the system may prompt input rules (i.e., the correct input format) near the corresponding data fields to reduce occurrence of data input errors. For example, a rule specifying "Input may not exceed 200 characters" may be displayed near an input field. Although a number of ways such error messages are displayed, some problems still remain in efficiently recollecting erroneous data from a user.

25

SUMMARY

[0005] One embodiment of the present invention provides a system for re-entering erroneously entered data fields in a web form. During operation, the system identifies one or more data fields with erroneous inputs in a first web form. Upon submission of the first web
30 form, the system restructures the identified data fields based on association information of a respective identified data field and displays the restructured data fields in a resubmission window distinct from the first web form.

[0006] In a variation of this embodiment, for identifying the one or more data fields, the system determines whether a data field is associated with an error message.

[0007] In a further variation, the system displays the error message with the associated data field in the resubmission window.

[0008] In a variation of this embodiment, the system identifies an associated data field for a first data field of the one or more data fields, obtains a current value for the associated data field, and displays a name of the associated data field and the current value with the first data field in the resubmission window.

[0009] In a further variation, the system avoids repetition of the associated data field and the current value for a second data field of the one or more data fields in the resubmission window. The associated data field is associated with the second data field.

[0010] In a variation of this embodiment, the system identifies categorization information for a first data field of the one or more data fields and displays the first data field in the resubmission window based on the categorization information.

[0011] In a variation of this embodiment, the system controls the resubmission window to constantly stay at a predetermined position on a screen displaying the resubmission window.

[0012] In a variation of this embodiment, the resubmission window is one of: (i) a floating div, (ii) a JavaScript popup window, and (iii) a second web form in a separate tab or window in a browser.

[0013] In a variation of this embodiment, upon submission of the first form, the system precludes a user of the first form from going back to the first form for a data input.

[0014] In a variation of this embodiment, the system orders the identified data fields in the resubmission window based on an order of the identified data fields in the first form.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] The accompanying drawings herein, which are incorporated herein and constitute a part of the specification, illustrate several exemplary embodiments of the present application and together with the description, serve to illustrate the present application, construing no limitation to the present application. In the drawings:

[0016] FIG. 1A illustrates an exemplary web form with erroneous inputs.

[0017] FIG. 1B illustrates an exemplary restructured data fields with erroneous inputs in a resubmission window, in accordance with an embodiment of the present application.

[0018] FIG. 2 illustrates an exemplary process of a data representation system generating a restructured resubmission window comprising data fields with erroneous inputs, in accordance with an embodiment of the present application.

[0019] FIG. 3 presents a flowchart illustrating the process of a data representation system generating a restructured resubmission window comprising data fields with erroneous inputs, in accordance with an embodiment of the present application.

[0020] FIG. 4A illustrates an exemplary association table comprising association (and
5 category) information of the data fields of a form, in accordance with an embodiment of the present application.

[0021] FIG. 4B illustrates an exemplary configuration table comprising configuration, value, and error information of the data fields of a form, in accordance with an embodiment of the present application.

10 [0022] FIG. 4C illustrates exemplary restructured data fields with erroneous inputs, in accordance with an embodiment of the present application.

[0023] FIG. 5A presents a flowchart illustrating the process of a data representation system restructuring the data fields with erroneous inputs, in accordance with an embodiment of the present application.

15 [0024] FIG. 5B presents a flowchart illustrating the process of a data representation system categorizing the data fields with erroneous inputs, in accordance with an embodiment of the present application.

[0025] FIG. 6 is a schematic structural diagram of an exemplary data representation system, in accordance with an embodiment of the present application.

20 [0026] In the figures, like reference numerals refer to the same figure elements.

DETAILED DESCRIPTION OF EMBODIMENTS

[0027] Embodiments of the present invention solve the problem of requesting a user, in an efficient and user-friendly manner, to re-enter erroneously entered data fields in a web form
25 by restructuring and displaying only those data fields. With existing technologies, when a system verifies the entered data and detects an input error, regardless of the number and locations of the erroneously entered data fields, the system still returns to the original form page to the user and requests the user to correct the mistakenly entered data fields. When locating and correcting the mistakenly entered data fields, the user would need to search for each such data field, until the
30 last one is found and corrected. In this process, the user might scroll down a long web form, and the system might need to display a large number of data fields, even though majority of these data fields have correct data. Often, the user might need to enter multiple commands (such as dragging a mouse, scrolling a wheel, etc.) to navigate through a long form and identify the data fields that need correction. This could significantly reduce the data processing efficiency and be
35 frustrating to the user to find the data fields that need correction.

[0028] To solve this problem, embodiments of the present invention provides a system that restructures the data fields with erroneous inputs in another resubmission window on the page displaying the form and displays only these data fields to a user. Examples of a resubmission window include, but are not limited to, a window or tab of a browser, a floating div, and a popup window. The system prompts the user of the context associated with these data fields, which indicate where the data fields were originally located. In this way, the user can be reminded of the correct content to be entered into these data fields. Furthermore, the system can also include a reminder of the correct data input format. For example, if a data field only accepts numeric digits and a user has provided an alphanumerical input, the system displays only that data field and reminds the user that the data field only accepts numeric inputs.

[0029] FIG. 1A illustrates an exemplary web form with erroneous inputs. As illustrated in FIG. 1A, a form 102 comprises a plurality of data fields. In some embodiments, form 102 is a HyperText Markup Language (HTML) form, which can be displayed in a browser. Form 102 can support both client-side scripting (e.g., JavaScript) and server-side scripting (e.g., Active Server Pages (ASP) and JavaServer Pages (JSP)). A user provides inputs to the corresponding data fields. For example, form 102 includes a data field 104 for a username of a user account. Data field 104 accepts an alphanumeric input from the user. In some embodiments, a data representation system 100 verifies the input (i.e., the entered data) from the user. System 100 can be hosting (e.g., a server) or displaying (e.g., a client) form 102.

[0030] Suppose that the input to data fields 104 (username), 106 (notes), and 108 (telephone) of form 102 include erroneous inputs. With existing technologies, when system 100 detects the errors, regardless of the number and locations of the erroneously entered data fields, system 100 still returns to original form 102 to the user and requests the user to correct the mistakenly entered data fields. For example, system 100 inserts an error message 110 indicating that data field 106 cannot be more that two hundred characters long and returns to form 102.

[0031] However, when locating and correcting data fields 104, 106, and 108, the user would need to search for each such data field in form 102, until the last one is found and corrected. If form 102 is a long web form, the user need to scroll down and check a large number of data fields, even though majority of these data fields may have correct data. Often, the user might need to enter multiple commands (such dragging a mouse, scrolling a wheel, etc.) to navigate through a long form and identify data fields 104, 106, and 108. This could significantly reduce the data processing efficiency and be frustrating to the user to find the data fields that need correction.

[0032] To solve this problem, system 100 can restructure data fields 104, 106, and 108, and display only these data fields to the user. FIG. 1B illustrates an exemplary restructured data

fields with erroneous inputs in a resubmission window, in accordance with an embodiment of the present application. In this example, system 100 displays these selected data fields 104, 106, and 108 in another resubmission window 150. It should be noted that displaying these selected data fields include both displaying the name of the data field and the corresponding input field (e.g., as they appear on form 102). Resubmission window 150 can be a window or tab (e.g., of a browser), or in a floating window (e.g., a JavaScript popup window or a floating div) on the page displaying form 102.

[0033] System 100 provides the user with the context associated with these data fields, which indicates where these data fields are originally located. In this way, the user can be reminded of the correct content to be entered into these data fields. In resubmission window 150, data fields 104, 106, and 108 appear in an order they have appeared in form 102. Furthermore, system 100 can add a reminder in resubmission window 150 indicating the correct input format of a respective data field. For example, if data field 108 (telephone) is empty, system 100 reminds the user that it's a required field (i.e., the input should not be empty).

[0034] Since resubmission window 150 can receive the previously erroneous user inputs of form 102, upon submission of form 102, system 100 can preclude the user from going back to form 102 for any data input. In other words, when the user submits form 102, system 100 only uses resubmission window 150 for any subsequent inputs from the user. Furthermore, if the user provides any erroneous input to a data field in resubmission window 150, system 100 can generate another resubmission window comprising that data field and precludes the user from going back to resubmission window 150 for an input. In this way, system 100 continues to use new resubmission windows for the data fields with erroneous inputs. It should be noted that the process of identifying data fields with erroneous inputs in resubmission window 150 can be similar to form 102. On the other hand, if resubmission window 150 is short (e.g., includes fewer data fields than a threshold), system 100 can continue to use resubmission window 150 for subsequent erroneous inputs.

[0035] In some embodiments, resubmission window 150 may be displayed as a "floating div" (e.g., an HTML floating div). To allow the user to easily concentrate on resubmission window 150, the floating div can be controlled to constantly stay at a predetermined position on the screen (e.g., a computer, cell phone, or tablet screen) currently displaying resubmission window 150. This predetermined position can be the center of the screen or defined based on other properties (e.g., the current screen size and/or type). The system can control the floating div to constantly stay at the predetermined position on the screen by calculating the relative position of the screen based on the movement of a pointing device. In some embodiments, the system determines the position based on the formula: vertical position of the scroll bar of the

current browser + (visible height of the browser window – visible height of the current object)/2.
For example, in JavaScript, the formula can be: `document.body.scrollTop + (document.body.clientHeight - this.clientHeight)/2`).

[0036] FIG. 2 illustrates an exemplary process of a data representation system generating
5 a restructured resubmission window comprising data fields with erroneous inputs, in accordance
with an embodiment of the present application. During operation, the system identifies the data
fields of form 102 (i.e., the original form) that have erroneous inputs (operation S202). The
system then determines a presentation style for presenting the identified data fields to the user
(operation S204). The system then restructures the identified data fields based on the determined
10 presentation style (e.g., association information of the data fields) (operation S206). The system
can also categorize the data fields based on the presentation style (e.g., categorization
information of the data fields). In this way, the system generates resubmission window 150 (i.e.,
the resubmission form) comprising data fields 104, 106, and 108 of form 102.

[0037] FIG. 3 presents a flowchart illustrating the process of a data representation system
15 generating a restructured resubmission window comprising data fields with erroneous inputs, in
accordance with an embodiment of the present application. During operation, the system detects
a form submission and traverses the data fields of the form (operation S301). The system
determines whether a data field of the form has one or more erroneous inputs (operation S302).
A erroneous input is caused during the data entering process of a user. The system then checks
20 whether an erroneous input exists for the data field (operation S303). If an erroneous input exists
for the data field, the system marks the data field as erroneous (operation S304). If an erroneous
input does not exist for the data field or the system has marked the data field, the system
determines whether all fields have been checked (operation S305). If all fields have not been
checked, the system determines whether the next data field of the form has an erroneous input
25 (operation S302).

[0038] If all fields have been checked, the system restructures the extracted data fields
(operation S306). In the example in FIGs. 1A and 1B, the system verifies the respective values
of the data fields in form 102. The system identifies data fields with erroneous inputs from form
102 based on the verification. The system then restructures and reorganizes the identified data
30 fields so that a resubmission window (e.g., resubmission window 150 in FIG. 1B) can be
constructed from the restructured data fields. During the restructuring process, the system can
order the identified data fields based on the order of the data fields in original form (e.g., form
102). Furthermore, the identified data fields can also be categorized based on a categorization of
the data fields.

[0039] The system then displays the restructured data fields in another resubmission window (operation S307), as described in conjunction with FIG. 1B. The resubmission window can be another form in a browser. The resubmission window can also be a floating window. The resubmission window includes the data fields with erroneous inputs from the original form (e.g., form 102). In this way, the user directly provides inputs in the resubmission window, thereby avoiding returning to the original form and searching the form for the data fields with erroneous inputs.

[0040] In some embodiments, the system can generate one or more tables to represent different aspects of the data fields of an original form. FIG. 4A illustrates an exemplary association table comprising association (and category) information of the data fields of a form, in accordance with an embodiment of the present application. A data representation system can generate an association table 400, which represents the association information of one or more data fields of an original form. For example, entries 402 and 404 of table 400 correspond to a customer's name and an account nickname of a customer (e.g., of an online shopping service). Table 400 indicates that a nickname of the account is associated with the customer name. Similarly, entries 406 and 408 of table 400 correspond to the account's contact name and a telephone number. Table 400 indicates that the telephone number is associated with the contact name.

[0041] In some embodiments, table 400 also includes category information of the data fields. It should be noted that the category information can also be in a separate table. This category information categorizes different data fields of a form. For example, entries 402 and 404 of table 400 can be categorized as a customer's basic information, and entries 406 and 408 of table 400 can be categorized as the customer's contact information. The association information in table 400 allows the system to obtain associated information of a data field with erroneous input in the resubmission window. Furthermore, the categorization information in table 400 allows the system to group the data fields with erroneous inputs based on their respective categories. In this way, the system provides a context of the data fields to the user, thereby providing the user with a better understanding of the correct inputs of the data fields.

[0042] FIG. 4B illustrates an exemplary configuration table comprising configuration, value, and error information of the data fields of a form, in accordance with an embodiment of the present application. A data representation system can generate a configuration table 410. Table 410 includes an identifier, name, type, length, current value, and error message, if any, for one or more data fields of the original form. In some embodiments, the system uses a div identifier as the identifier of a data field. For example, the system uses identifiers 412, 414, 416, and 418 for data fields "customer name," "nickname," "contact name," and "telephone,"

respectively. To identify the data fields with erroneous inputs, the system can traverse the entries (i.e., the rows) of table 410 and checks whether an entry has an error message. Upon identifying such an entry, the system can use the identifier of that entry for subsequent references to the data field. For example, the system can use the identifier to extract the data field from the original form and insert into the resubmission window.

[0043] FIG. 4C illustrates exemplary restructured data fields with erroneous inputs, in accordance with an embodiment of the present application. Suppose that, in the examples in FIGs. 4A and 4B, data fields “nickname” and “telephone” have erroneous inputs. A data representation system obtains association information of these data fields by determining that data fields “nickname” and “telephone” are associated with data fields “customer name” and “contact name,” respectively. Data fields “customer name” and “contact name” can be referred to as the associated data fields for data fields “nickname” and “telephone,” respectively. The system then obtains the current values “XYZ” and “ABC” of “customer name” and “contact name,” respectively, from table 410. The system inserts the names of the associated data fields, as obtained from table 400, and corresponding current values, as obtained from table 410, before the corresponding data fields with erroneous inputs.

[0044] In the example in FIG. 4C, the system generates a resubmission window 450 for data fields “nickname” and “telephone.” The system inserts associated data field name “customer name” and its current value “XYZ” before (or after) “nickname” in resubmission window 450. The system also inserts associated data field name “contact name” and its current value “ABC” before (or after) “telephone.” In this way, the system provides a context for the data fields with erroneous inputs. In some embodiments, the system also obtains the error message from table 410 for data fields “nickname” and “telephone,” and inserts the messages next to the corresponding data fields. The system can obtain the error messages based on identifiers 414 and 418 of data fields “nickname” and “telephone,” respectively, from table 410.

[0045] In some embodiments, the system categorizes the data fields in resubmission window 450 based on their respective categories from table 400. For example, the system can obtain the categories for entries 402, 404, 406, and 408 (i.e., for the data fields with erroneous inputs and their associated data fields). The system then categorizes the data fields in resubmission window 450. For example, since the category for data fields “customer name” and “nickname” is “basic information of customer,” the system groups these data fields together in resubmission window 450. Similarly, since the category for data fields “contact name” and “telephone” is “contact information of customer,” the system groups these data fields together in resubmission window 450. The system can also insert the corresponding category for a respective group. In this way, in resubmission window 450, the system restructures and

categorizes the data fields based on the presentation styles described in conjunction with FIGs 4A and 4B.

[0046] FIG. 5A presents a flowchart illustrating the process of a data representation system restructuring the data fields with one or more erroneous inputs, in accordance with an embodiment of the present application. During operation, the system obtains association information associated with a respective data field with erroneous inputs (e.g., from an association table) (operation S501). The system then identifies the associated data fields from the association information (operation S502) and obtains the respective current values of the associated data fields (e.g., from a configuration table) (operation S503). Since a plurality of data fields can have the same associated data field, the system groups the data fields with erroneous inputs based on the corresponding associated data field in a resubmission window (operation S504). The system then inserts the associated data field names and their corresponding values before the corresponding group in the resubmission window (operation S505). This prevents repetition of an associated data field.

[0047] It should be noted that the system may repeat an associated data field in the resubmission window. The system can also remove the repeated associated data fields from a resubmission window. Under such circumstances, the system may not group the data fields with erroneous inputs and insets data fields individually in the resubmission window. In some embodiments, the system can obtain the error messages for the data fields with erroneous inputs and inserts the obtained error messages next to the corresponding data fields in the resubmission window (operation S506), as described in conjunction with FIG. 4C.

[0048] FIG. 5B presents a flowchart illustrating the process of a data representation system categorizing the data fields with erroneous inputs, in accordance with an embodiment of the present application. During operation, the system obtains categorization information associated with a respective data field with erroneous inputs (operation S521), as described in conjunction with FIG. 4A. The system then categorizes the data fields with erroneous inputs based on the categorization information in the resubmission window (operation S522), as described in conjunction with FIG. 4C. In some embodiments, the system also includes the categories in the resubmission window next to a respective category of the data fields.

Exemplary Apparatus

[0049] FIG. 6 is a schematic structural diagram of an exemplary data representation system, in accordance with an embodiment of the present application. A data representation system 600 can be used for re-entering erroneously entered data fields in a web form by restructuring and displaying only those data fields. System 600 includes a processor 610, a

memory 620, and a storage device 630. Storage device 630 typically stores instructions that can be loaded into memory 620 and executed by processor 610 to perform the methods described above. In one embodiment, the instructions in storage 630 can implement a determination module 632, a restructuring module 634, a categorization module 636, and a display module 638, all of which can communication with each other through various means.

[0050] In some embodiments, modules 632, 634, 636, and 638 can be partially or entirely implemented in hardware and can be part of processor 610. Further, in some embodiments, the system may not include a separate processor and memory. Instead, in addition to performing their specific tasks, modules 632, 634, 636, and 638, either separately or in concert, may be part of special-purpose computation engines.

[0051] Storage 630 stores programs to be executed by processor 610. Specifically, storage 630 stores a program that implements a system (application) for re-enter data fields in a form with erroneous inputs. During operation, the application program can be loaded from storage 630 into memory 620 and executed by processor 610. As a result, system 600 can perform the functions described above. System 600 can be further coupled to an optional display 680, a keyboard 660, and a pointing device 670, and can be coupled via one or more network interfaces to a network 682.

[0052] During operation, determination module 632 determines the data fields with erroneous inputs from a web form, as described in conjunction with FIG. 3. Restructuring module 634 restructures the data fields in a resubmission window, as described in conjunction with FIG. 5A. Categorization module 636 categorizes the data fields in a resubmission window, as described in conjunction with FIG. 5B. Display module 638 controls the resubmission window and displays the restructured and categorized data fields in the resubmission window, as described in conjunction with FIGs. 1B and 4C.

[0053] Embodiments of the present invention may be implemented on various universal or dedicated computer system environments or configurations. For example, such computer systems may include personal computers, server computers, handheld or portable devices, tablet-type devices, multiprocessor systems, microprocessor-based systems, set-top boxes, programmable electronic consumption devices, network PCs, minicomputers, mainframe computers, distributed computing environments including any of the above systems or devices, and the like.

[0054] Embodiments of the present invention may be described within the general context of computer-executable instructions executed by a computer, such as a program module. Generally, the program module includes a routine, a program, an object, an assembly, a data structure and the like for implementing particular tasks or achieving particular abstract data

types. Embodiments of the present invention may also be implemented in distributed computing environments, in which tasks are performed by remote processing devices connected via a communication network. In the distributed computing environments, program modules may be located in local and remote computer storage media that may include a storage device.

5 **[0055]** The data structures and computer instructions described in this detailed description are typically stored on a computer-readable storage medium, which may be any device or medium that can store code and/or data for use by a computer system. The computer-readable storage medium includes, but is not limited to, volatile memory, non-volatile memory, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs),
10 DVDs (digital versatile discs or digital video discs), or other media capable of storing computer-readable media now known or later developed.

[0056] The methods and processes described in the detailed description section can be embodied as code and/or data, which can be stored in a computer-readable storage medium as described above. When a computer system reads and executes the code and/or data stored on the
15 computer-readable storage medium, the computer system performs the methods and processes embodied as data structures and code and stored within the computer-readable storage medium.

[0057] Furthermore, methods and processes described herein can be included in hardware modules or apparatus. These modules or apparatus may include, but are not limited to, an application-specific integrated circuit (ASIC) chip, a field-programmable gate array (FPGA), a
20 dedicated or shared processor that executes a particular software module or a piece of code at a particular time, and/or other programmable-logic devices now known or later developed. When the hardware modules or apparatus are activated, they perform the methods and processes included within them.

[0058] The above description is presented to enable any person skilled in the art to make
25 and use the embodiments, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present disclosure. Thus, the present invention is not limited to the embodiments shown, but is to be
30 accorded the widest scope consistent with the principles and features disclosed herein.

What Is Claimed Is:

1. A computer-implemented method for re-entering erroneously entered data fields in a web form, comprising:

5 identifying, by a computing device, one or more data fields with erroneous inputs in the web form;

restructuring the identified data fields based on an order of the identified data fields in the web form; and

10 displaying the restructured data fields in a resubmission window distinct from the web form without displaying a data field of the web form with a correct input, wherein the resubmission window allows a user to reenter inputs into the restructured data fields.

2. The method of claim 1, wherein identifying the one or more data fields comprises determining whether a first data field of the one or more data fields is associated with an error message.

15 3. The method of claim 2, further comprising displaying the error message with the first data field in the resubmission window.

4. The method of claim 1, further comprising:
identifying a second data field associated with a first data field of the one or more data fields;

20 obtaining a current value for the second data field; and

displaying, with the first data field, a name and the current value of the second data field in the resubmission window.

5. The method of claim 4, further comprising avoiding repetition of the name and the current value of the second data field for a third data field of the one or more data fields in the
25 resubmission window, wherein the second data field is further associated with the third data field.

6. The method of claim 1, further comprising:
identifying categorization information for a first data field of the one or more data fields; and

30 displaying the first data field in the resubmission window based on the categorization information.

7. The method of claim 1, further comprising controlling the resubmission window to constantly stay at a predetermined position on a screen displaying the resubmission window.

5 8. The method of claim 1, wherein the resubmission window is one of:
a floating div;
a JavaScript popup window; and
a second web form in a separate tab or window in a browser.

10 9. The method of claim 1, further comprising, upon submission of the web form, precluding the user of the web form from going back to the web form for a data input.

10. The method of claim 1, further comprising ordering the identified data fields in the resubmission window based on the order of the identified data fields in the web form.

15

11. A non-transitory computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for receiving password input from a user, the method comprising:

20 identifying, by the computer, one or more data fields with erroneous inputs in a web form;

restructuring the identified data fields based on an order of the identified data fields in the web form; and

25 displaying the restructured data fields in a resubmission window distinct from the web form without displaying a data field of the web form with a correct input, wherein the resubmission window allows the user to reenter inputs into the restructured data fields.

12. The non-transitory computer-readable storage medium of claim 11, wherein identifying the one or more data fields comprises determining whether a first data field of the one or more data fields is associated with an error message.

30 13. The non-transitory computer-readable storage medium of claim 12, wherein the method further comprises displaying the error message with the first data field in the resubmission window.

14. The non-transitory computer-readable storage medium of claim 11, wherein the

method further comprises:

identifying a second data field associated with a first data field of the one or more data fields;

obtaining a current value for the second data field; and

5 displaying, with the first data field, a name and the current value of the second data field in the resubmission window.

15 15. The non-transitory computer-readable storage medium of claim 14, wherein the method further comprises avoiding repetition of the name and the current value of the second data field for a third data field of the one or more data fields in the resubmission window, wherein the second data field is further associated with the third data field.

16. The non-transitory computer-readable storage medium of claim 11, wherein the method further comprises:

identifying categorization information for a first data field of the one or more data fields; and

15 displaying the first data field in the resubmission window based on the categorization information.

17. The non-transitory computer-readable storage medium of claim 11, wherein the method further comprises controlling the resubmission window to constantly stay at a
20 predetermined position on a screen displaying the resubmission window.

18. The non-transitory computer-readable storage medium of claim 11, wherein the resubmission window is one of:

a floating div;

25 a JavaScript popup window; and

a second web form in a separate tab or window in a browser.

19. The non-transitory computer-readable storage medium of claim 11, wherein the method further comprises, upon submission of the web form, precluding the user of the web
30 form from going back to the web form for a data input.

20. The non-transitory computer-readable storage medium of claim 11, wherein the method further comprises ordering the identified data fields in the resubmission window based on the order of the identified data fields in the web form.

- 5 21. A computing system for facilitating dynamic load-based merging, the computing system comprising:
- a processor;
 - a memory coupled to the processor and storing instructions, which when executed by the processor cause the processor to perform a method, the method comprising:
- 10 identifying, by the processor, one or more data fields with erroneous inputs in a web form;
- restructuring the identified data fields based on an order of the identified data fields in the web form; and
 - displaying the restructured data fields in a resubmission window distinct from the web
- 15 form without displaying a data field of the web form with a correct input, wherein the resubmission window allows a user to reenter inputs into the restructured data fields.