



US 20220245328A1

(19) United States

(12) Patent Application Publication  
TSABBA

(10) Pub. No.: US 2022/0245328 A1  
(43) Pub. Date: Aug. 4, 2022

(54) CUSTOMIZABLE DATA MANAGEMENT  
FORM BUILDER METHOD AND DEVICES

(71) Applicant: BINYAMIN TSABBA, ELYACHIN  
(IL)

(72) Inventor: BINYAMIN TSABBA, ELYACHIN  
(IL)

(21) Appl. No.: 17/726,501

(22) Filed: Apr. 21, 2022

#### Related U.S. Application Data

(63) Continuation of application No. 16/214,045, filed on Dec. 8, 2018.

#### Publication Classification

(51) Int. Cl.

G06F 40/14 (2006.01)  
G06F 3/0486 (2006.01)  
G06F 3/16 (2006.01)  
G10L 17/00 (2006.01)

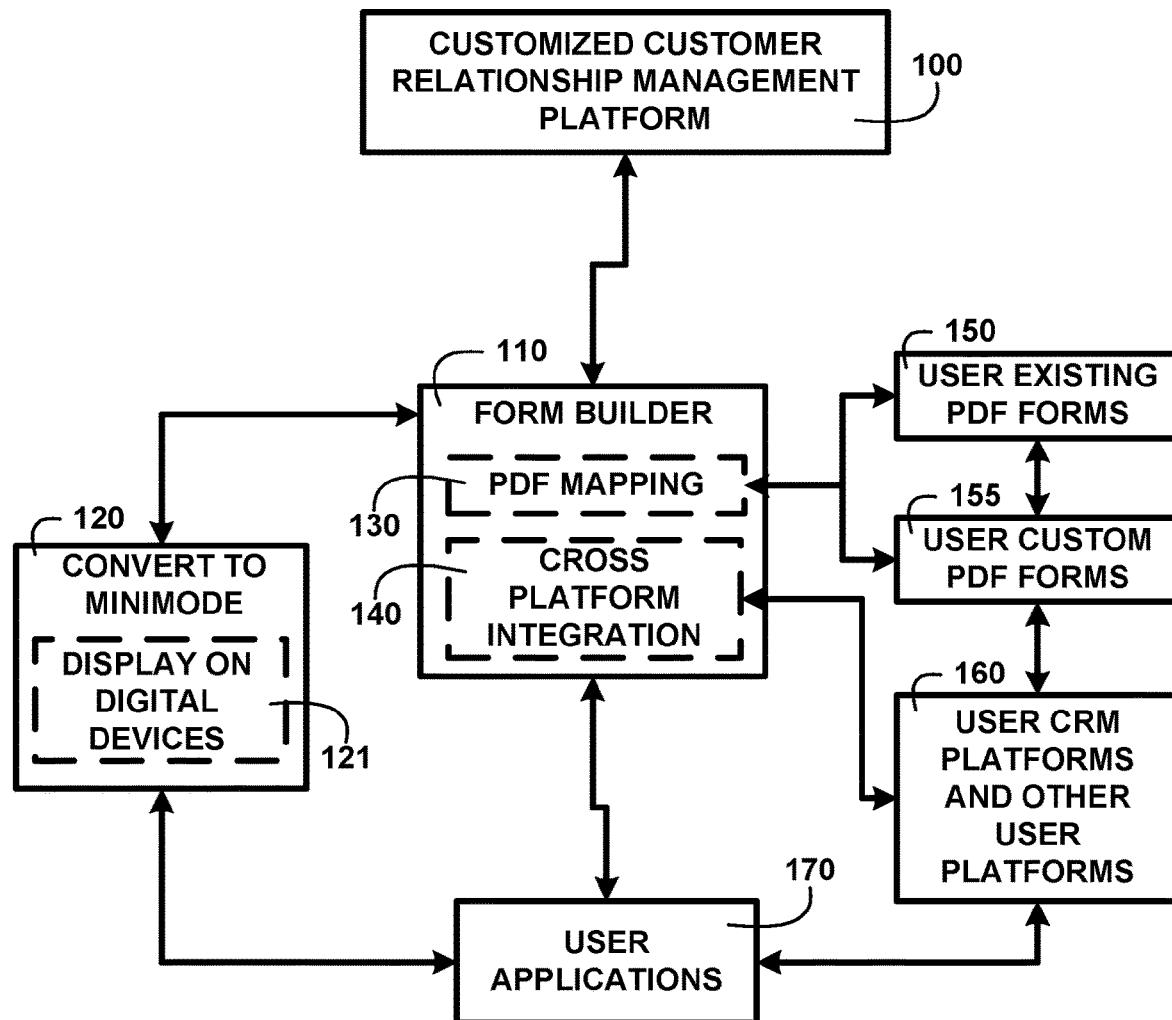
(52) U.S. Cl.

CPC ..... G06F 40/14 (2020.01); G10L 17/00  
(2013.01); G06F 3/167 (2013.01); G06F  
3/0486 (2013.01)

(57)

#### ABSTRACT

The embodiments disclose a method including providing a website interface within the customizable application builder with no code visualization tools for an application creation user, including a form builder for creating at least one form with at least one form element for receiving form input from an end user, each associated with a conditional rule builder, an auto pdf mapping tool, a manual pdf mapping tool and at least two dynamic pdf mapping tools, wherein upon submission of the form, the auto pdf mapping tool automatically generates a pdf of the entire form, including the at least one form element and the form input from the end user and wherein the end user can interactively integrate the entire auto pdf mapping tool automatically generated pdf form into at least one 3rd party application platform and access data from all of their 3rd party application platform using the pdf mapped elements.



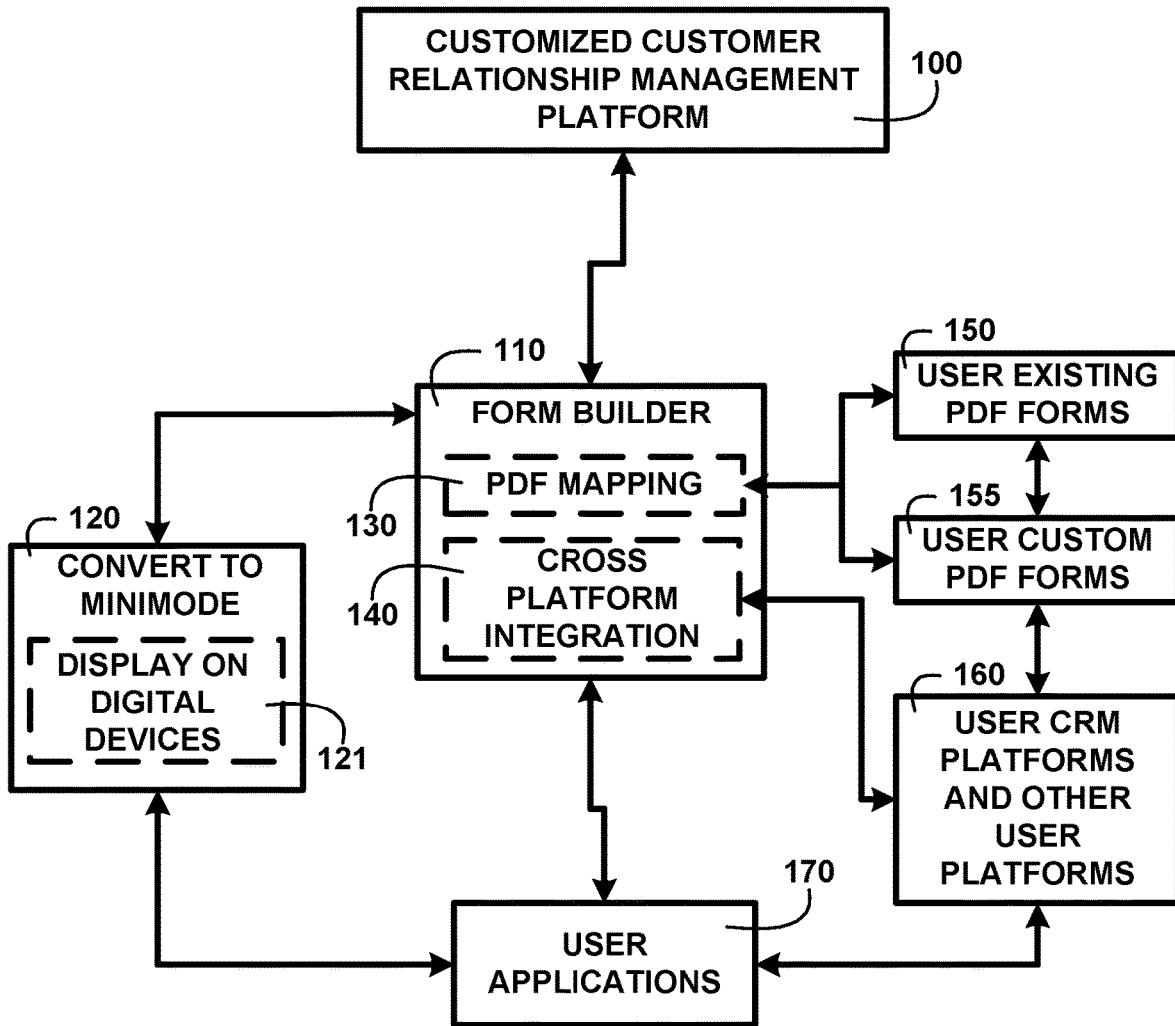
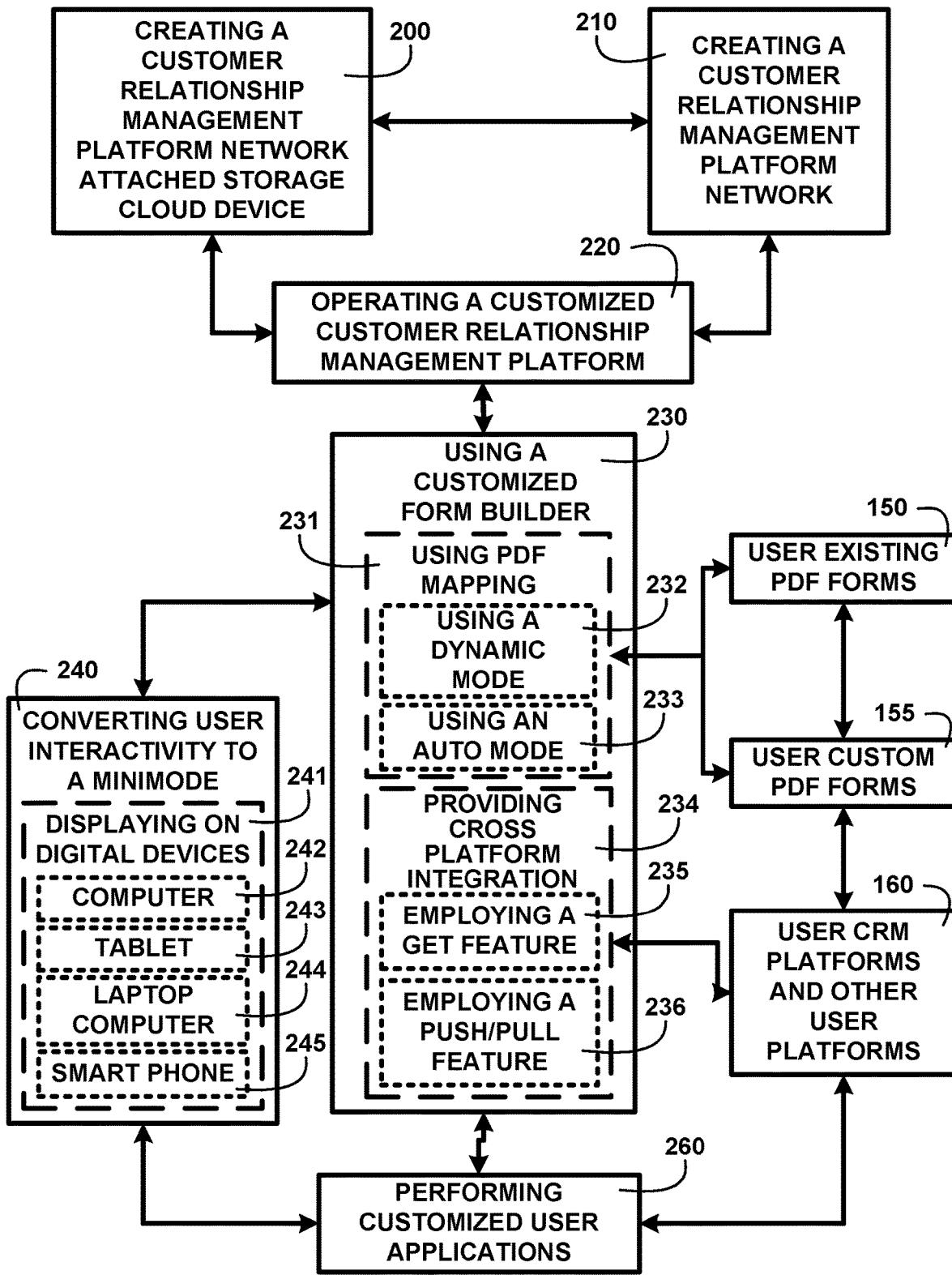


FIG. 1

**FIG. 2**

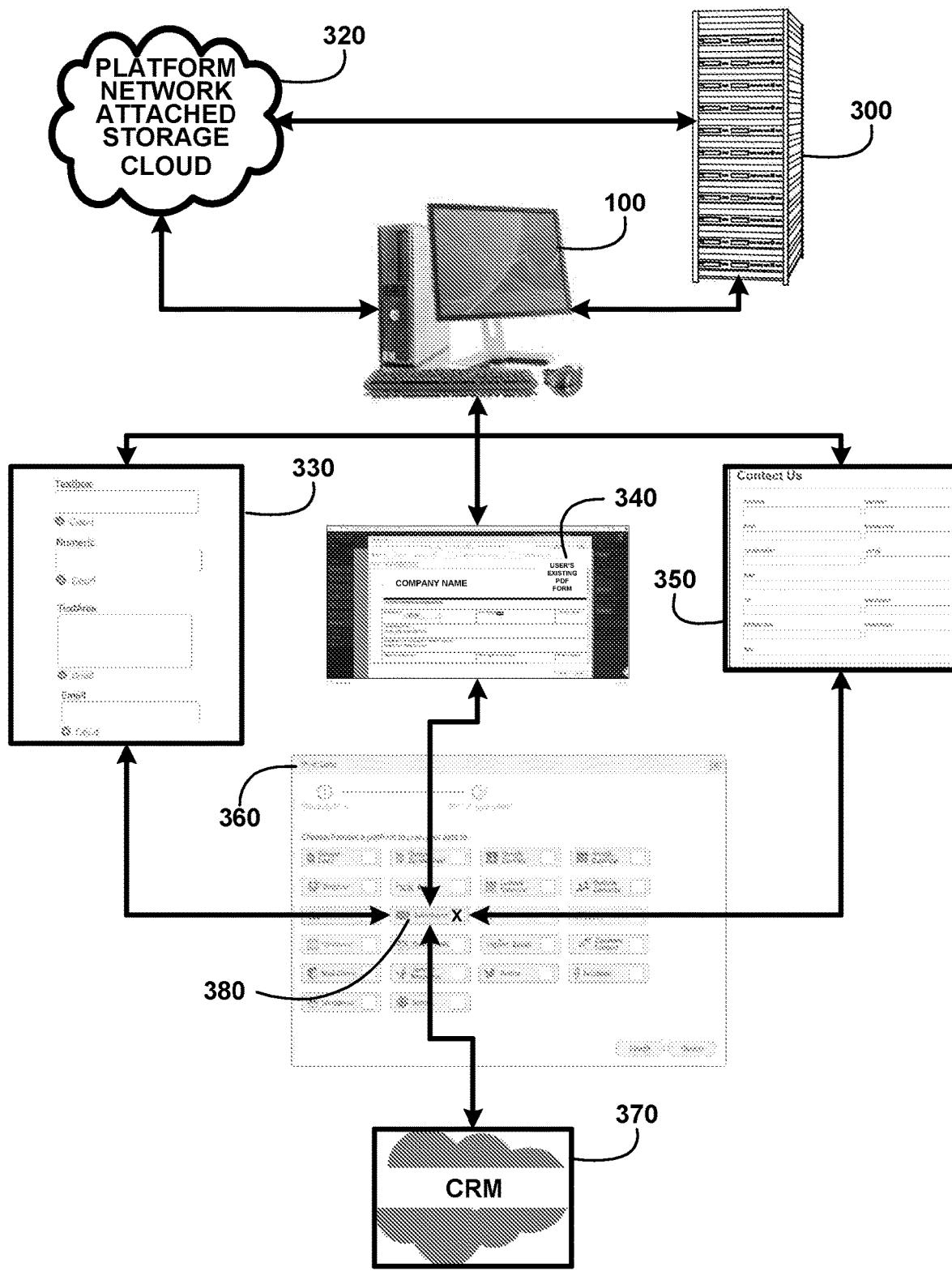


FIG. 3

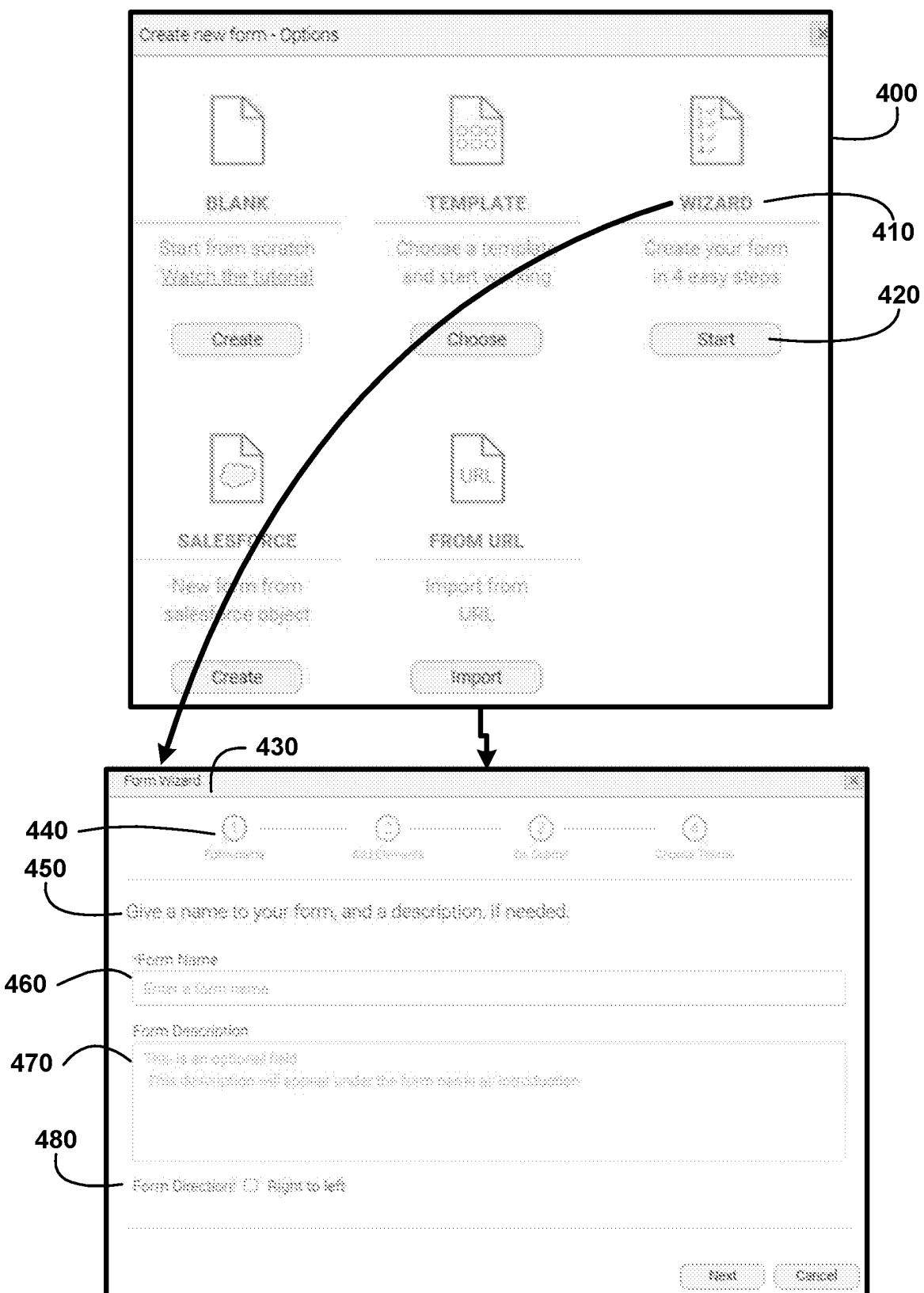
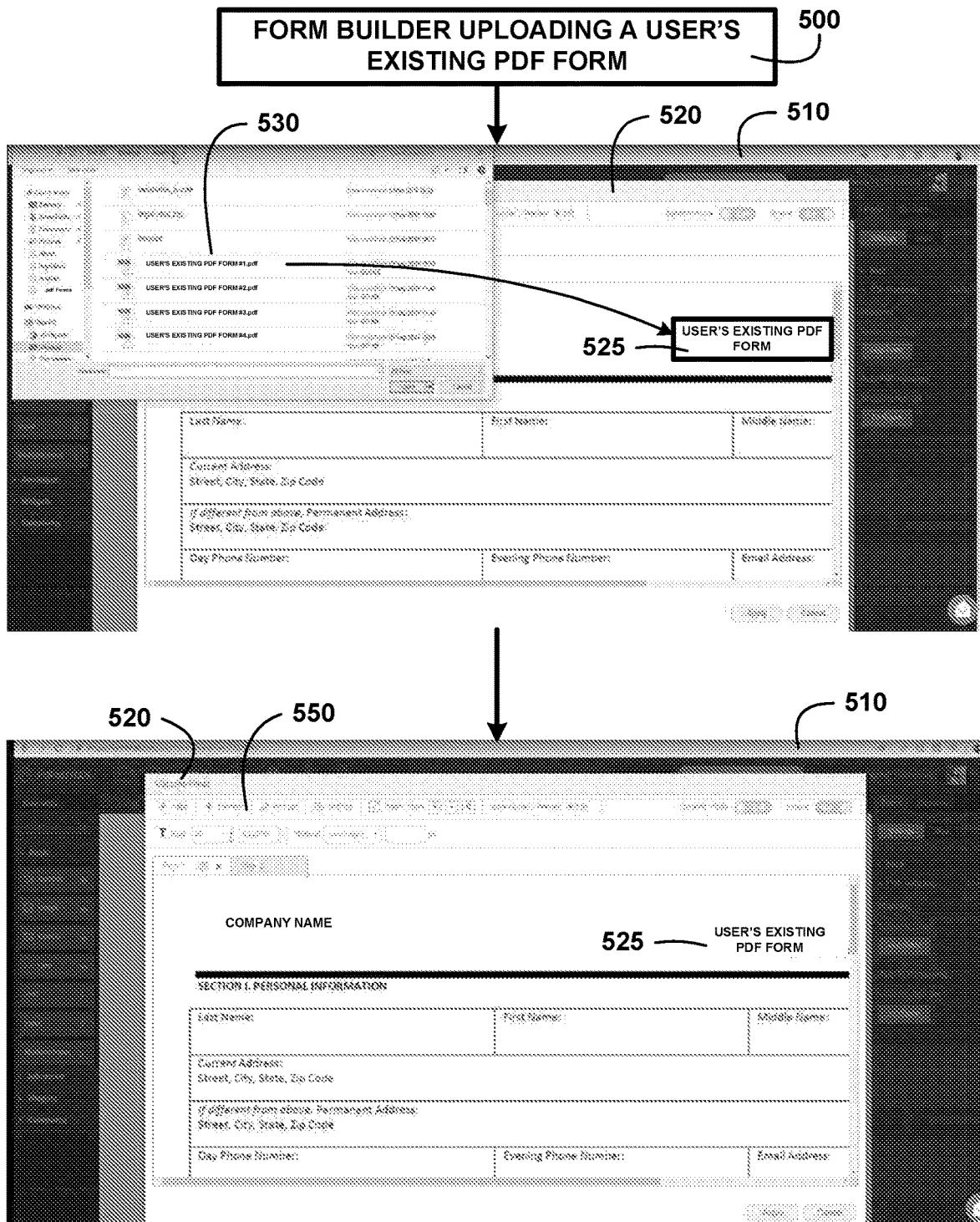
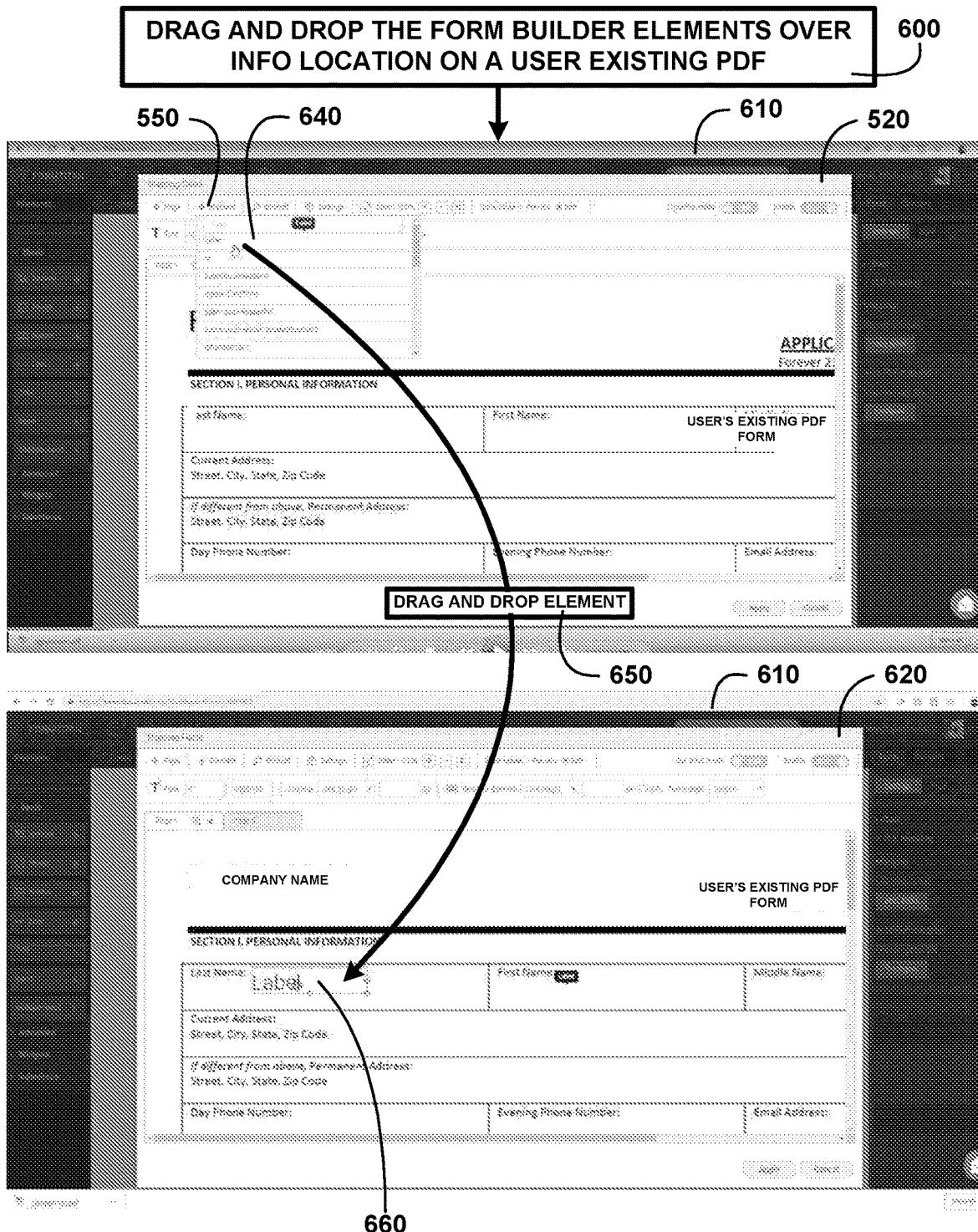


FIG. 4



**FIG. 5**



**FIG. 6**

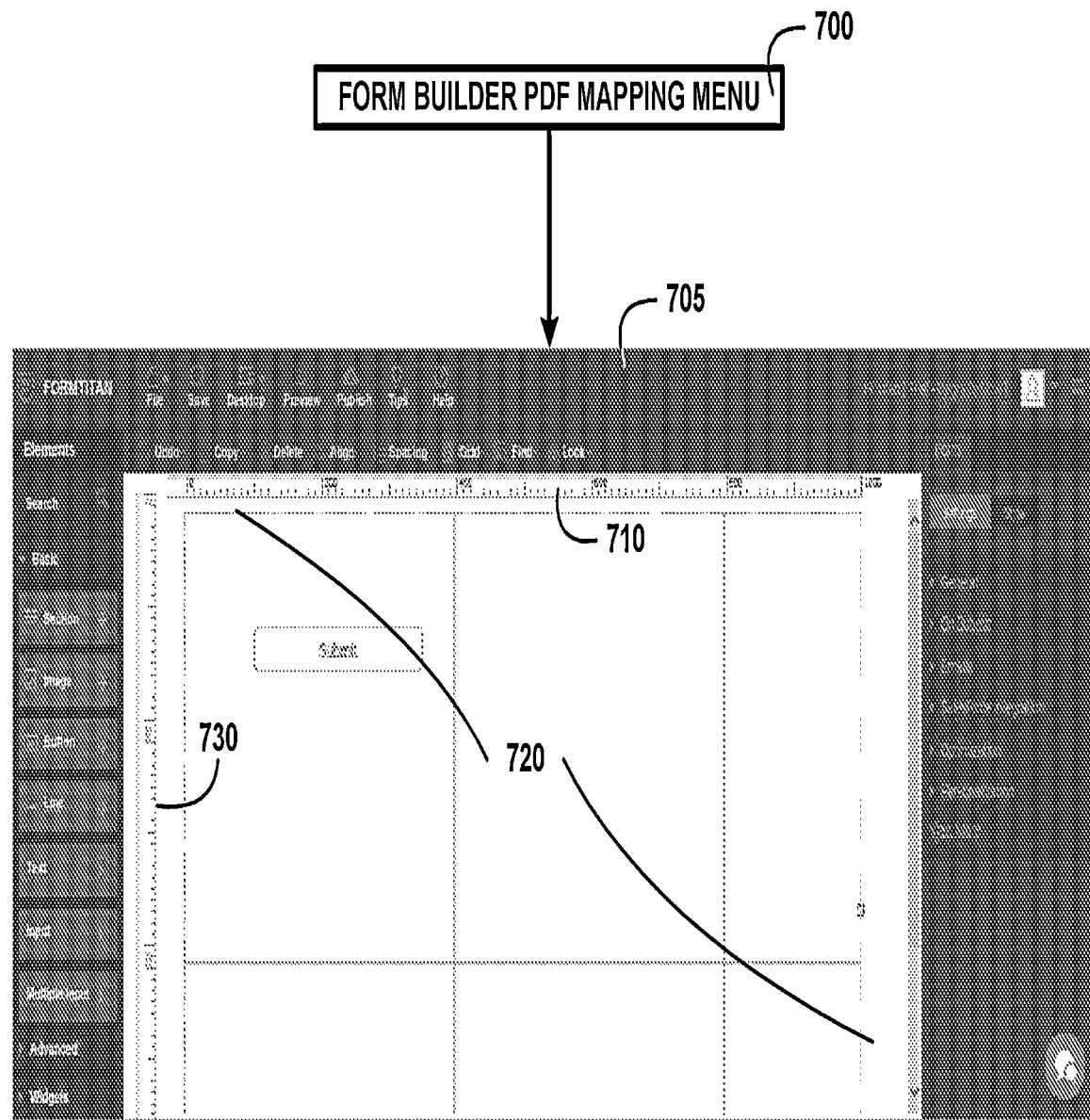
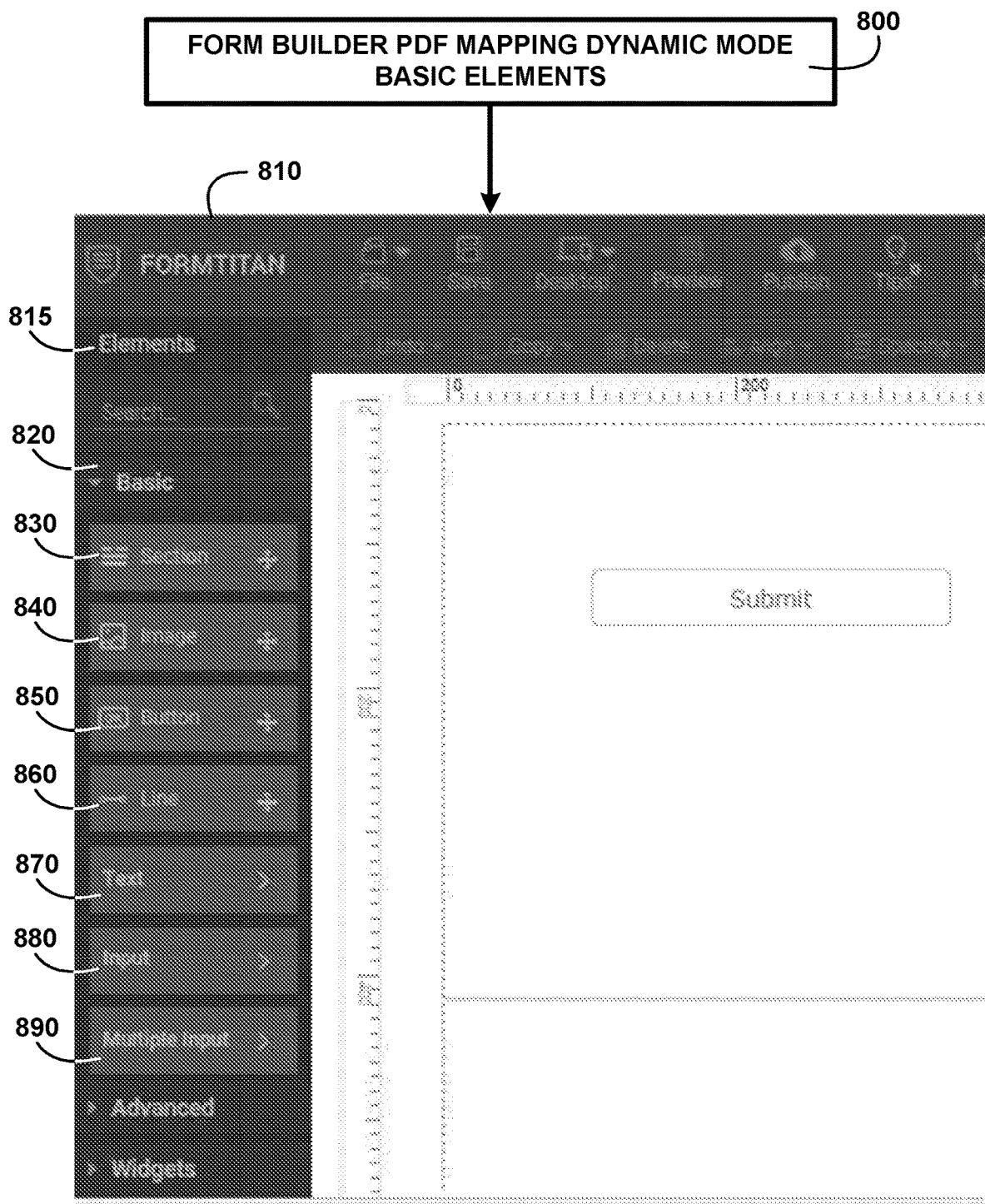
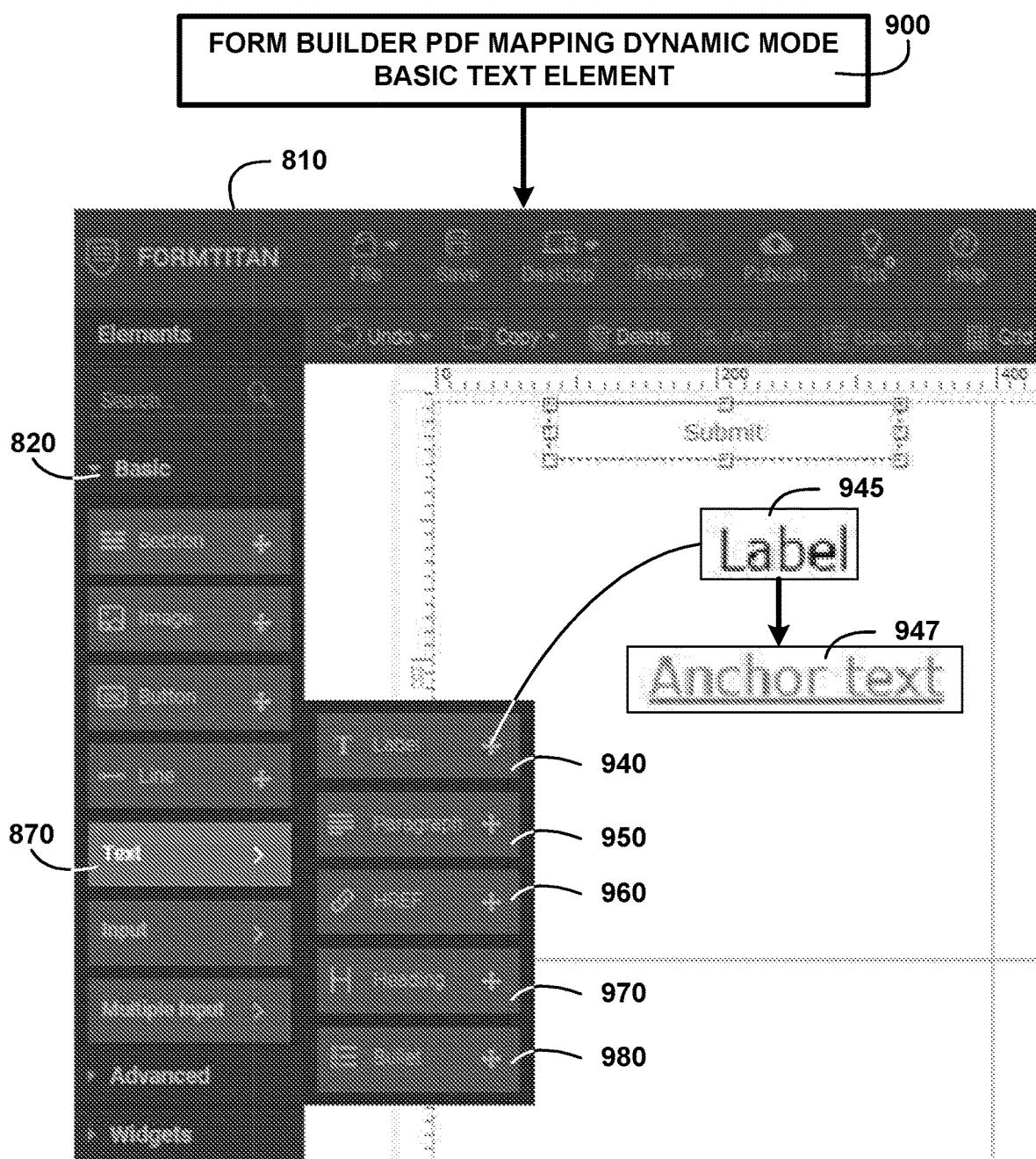


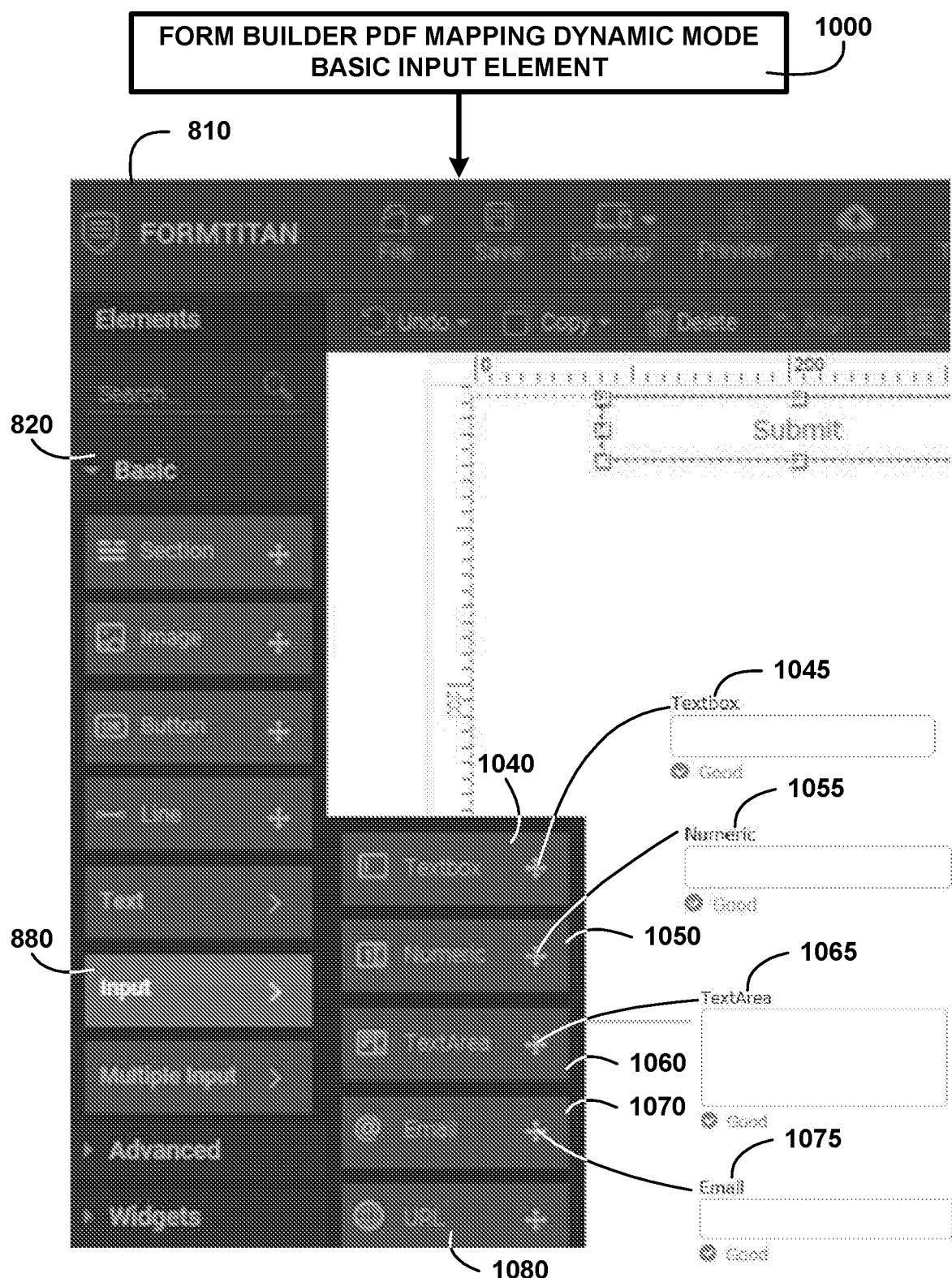
FIG. 7



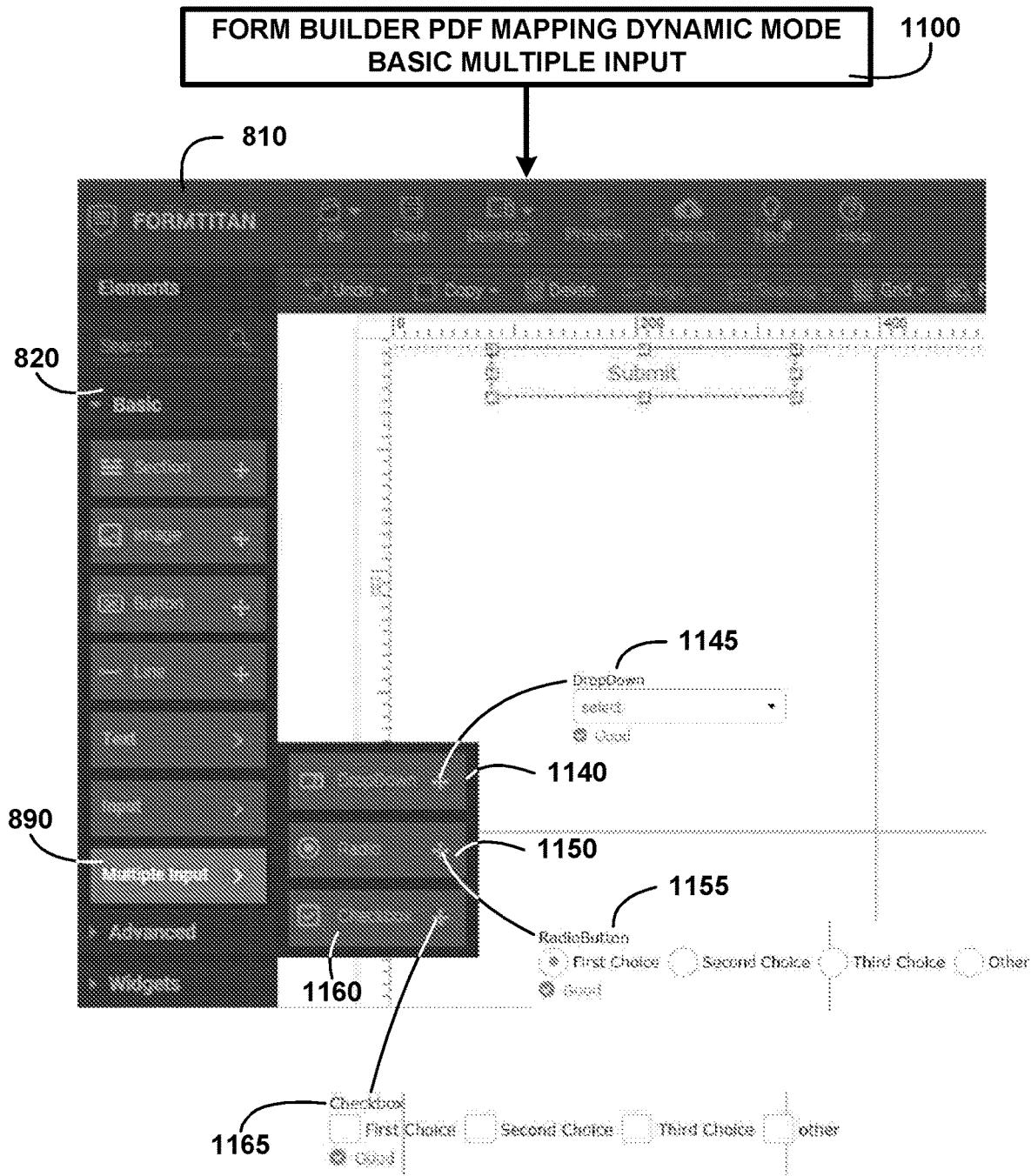
**FIG. 8**



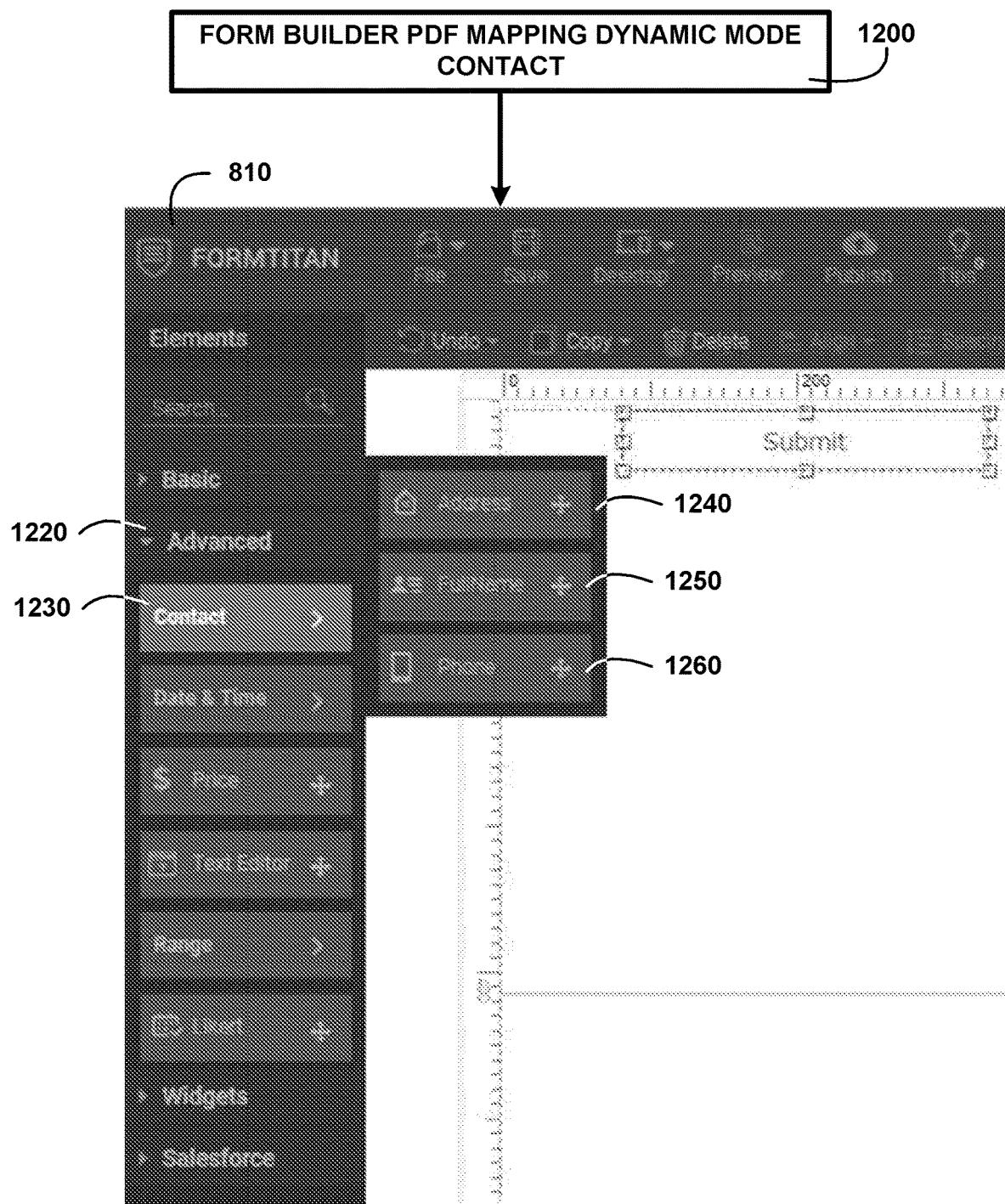
**FIG. 9**



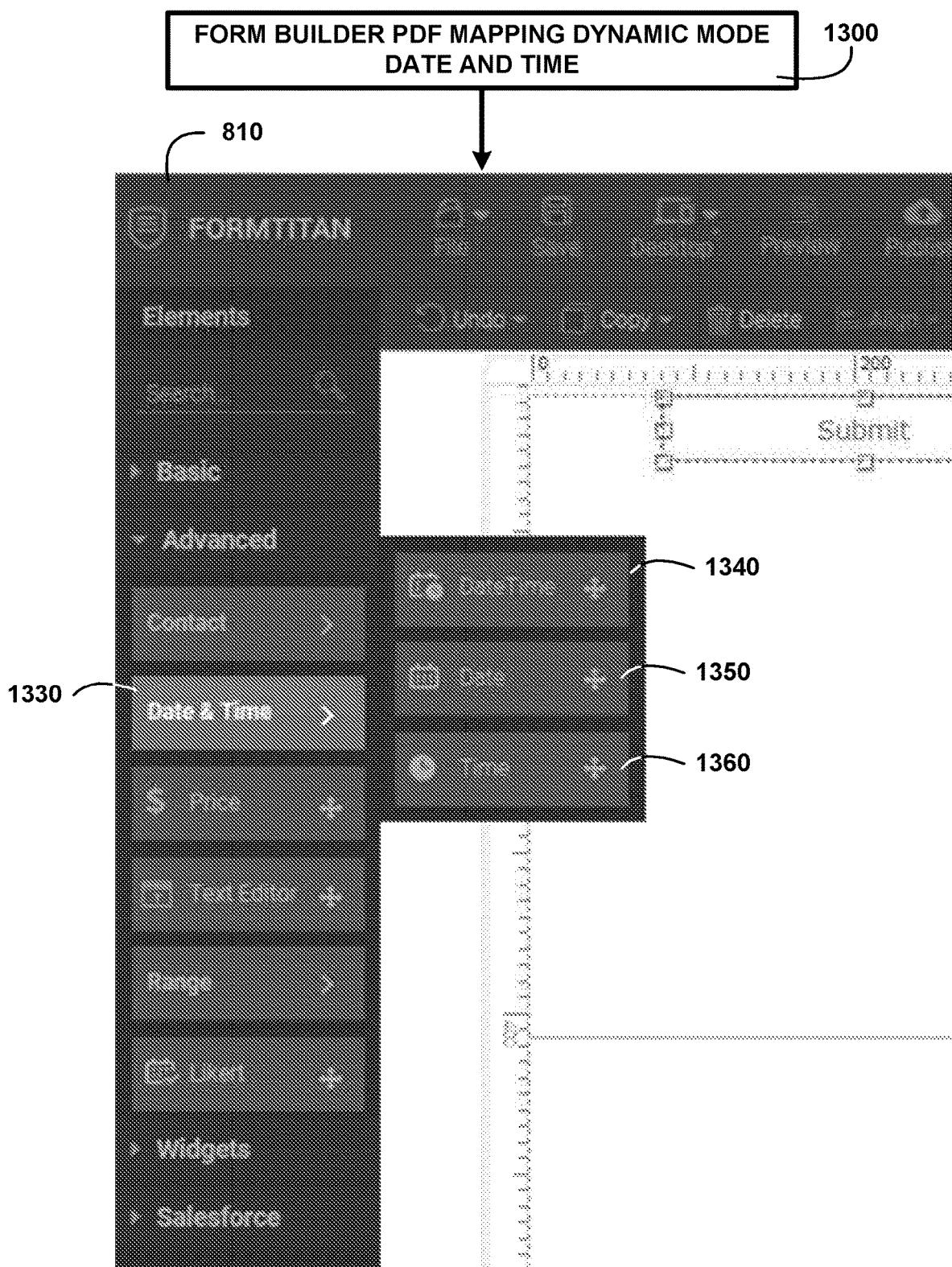
**FIG. 10**



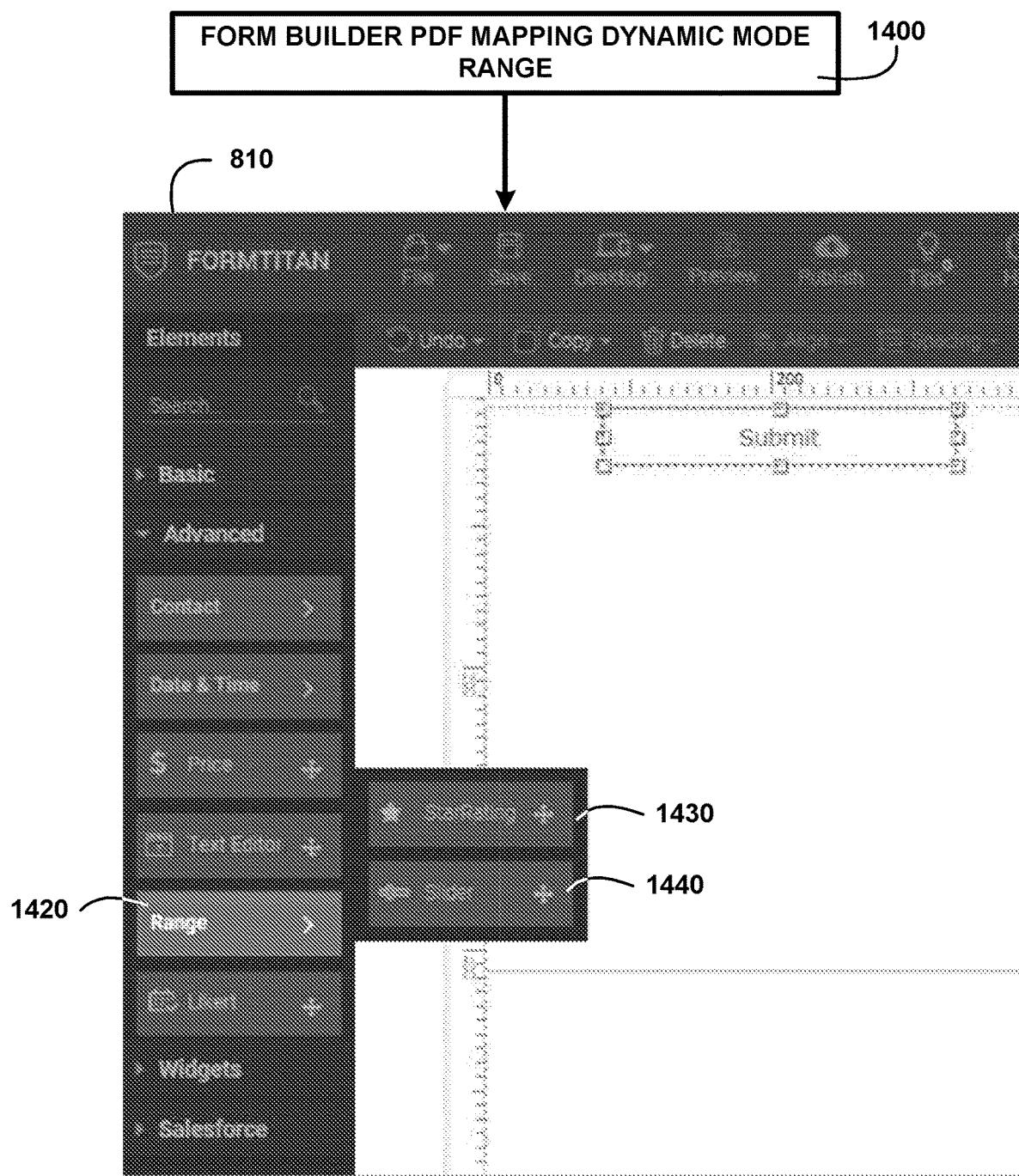
**FIG. 11**



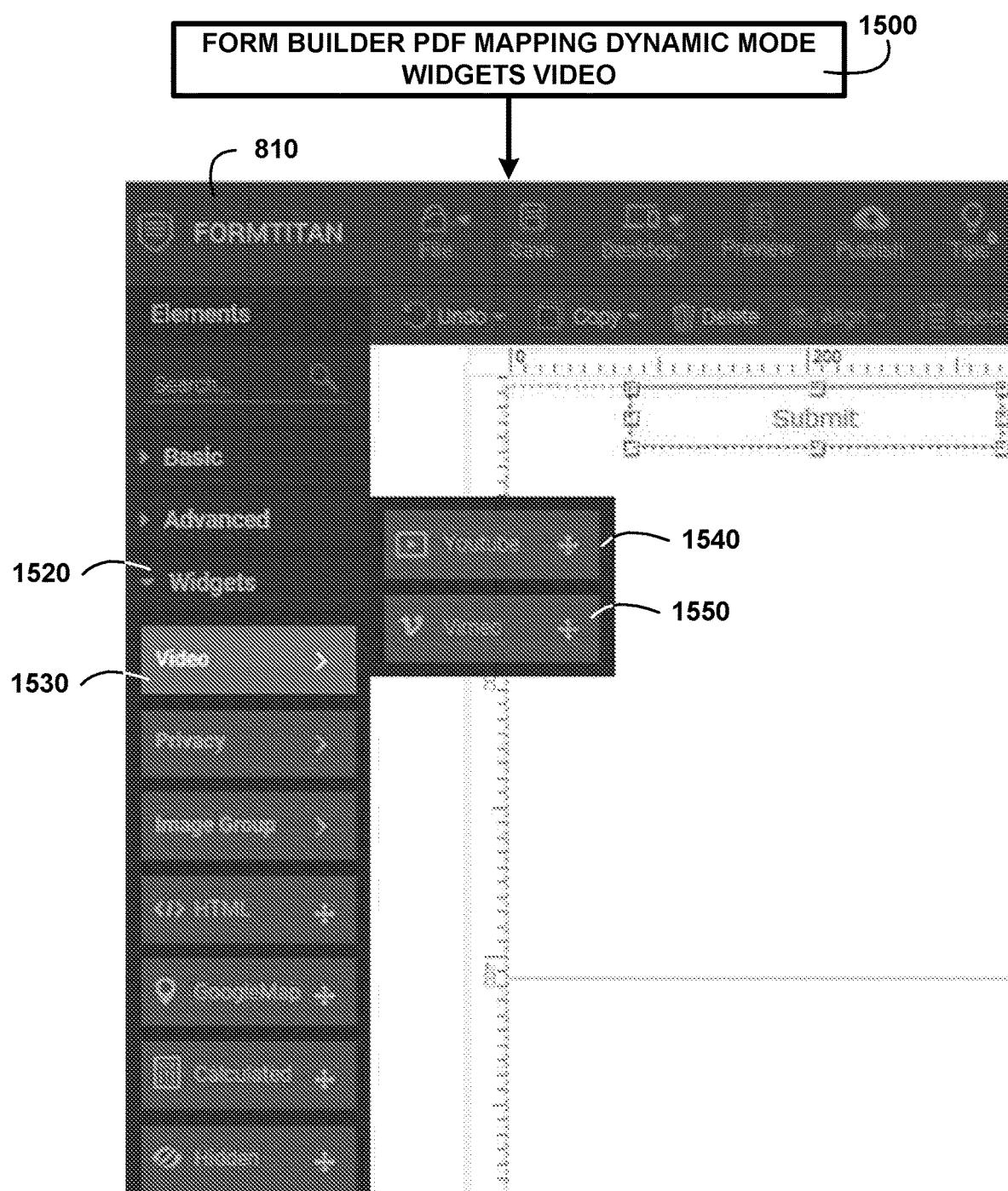
**FIG. 12**



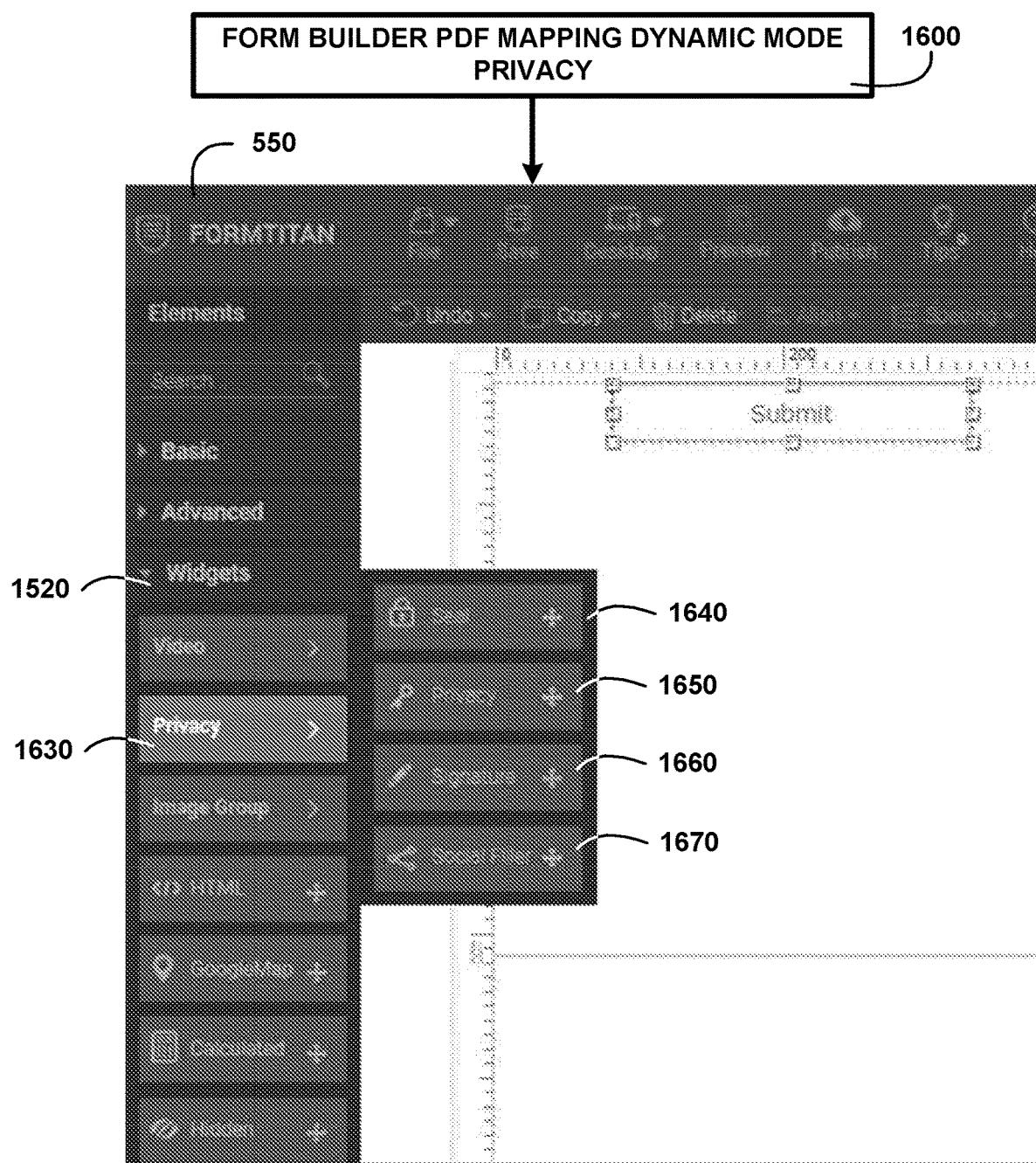
**FIG. 13**



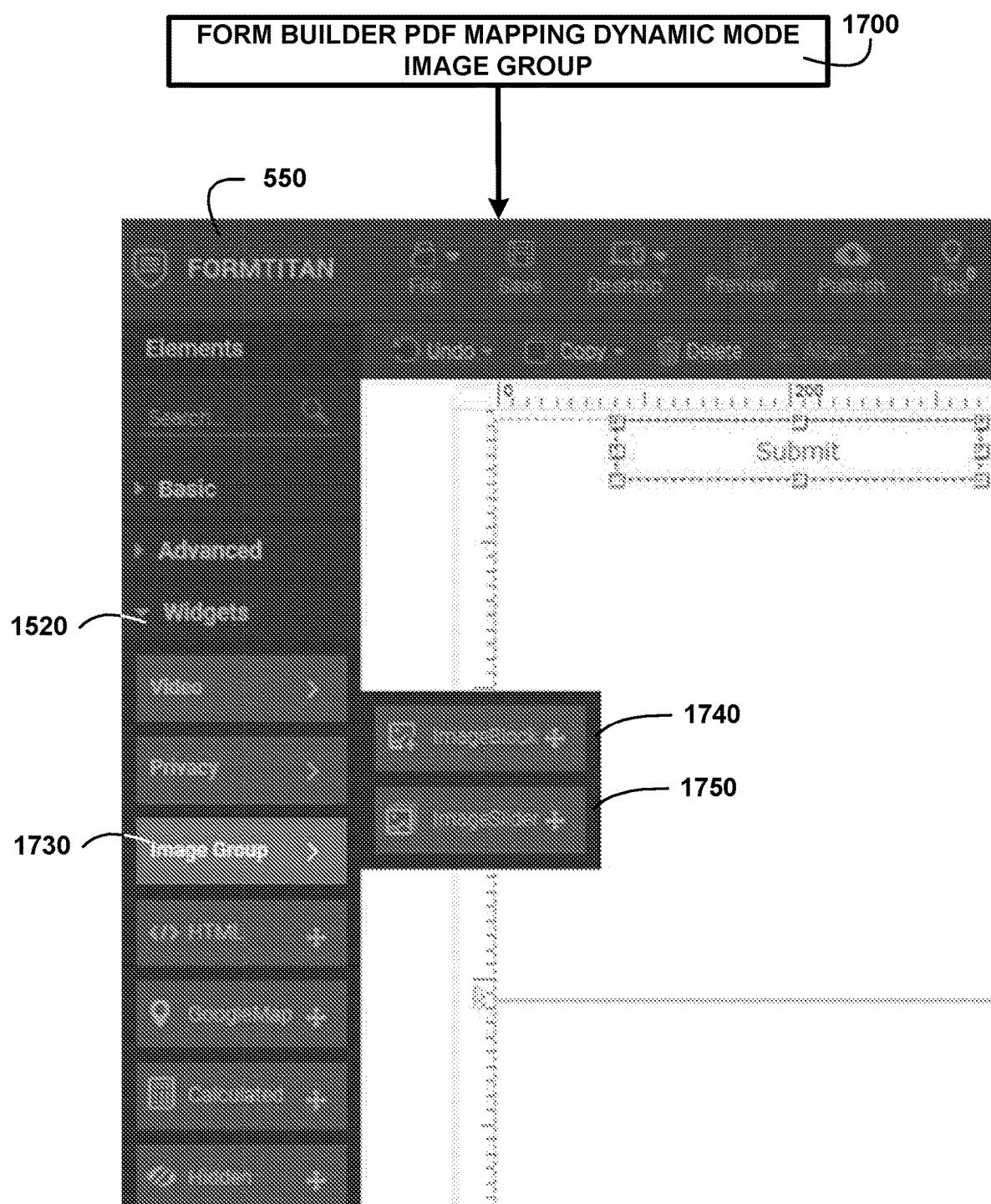
**FIG. 14**



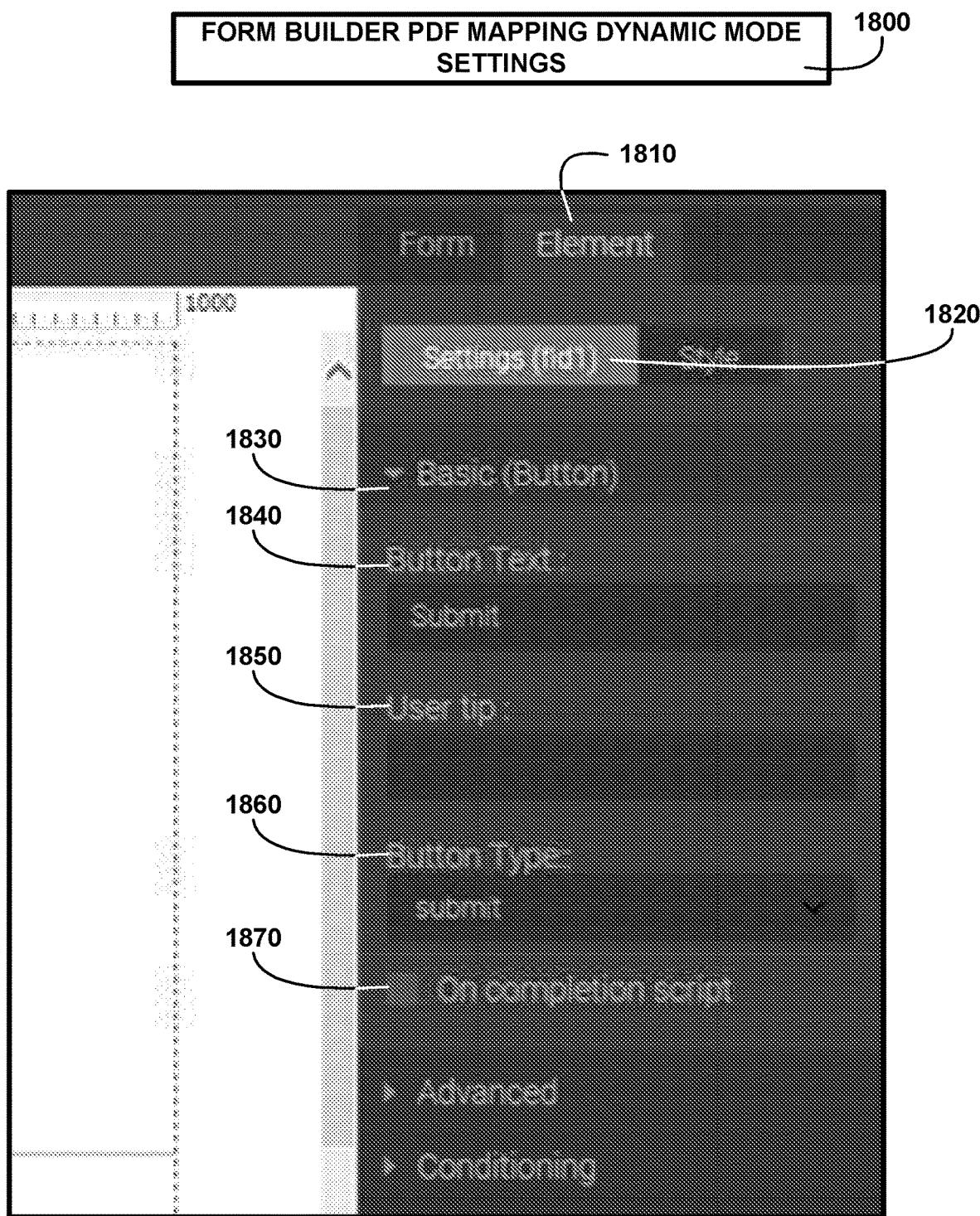
**FIG. 15**



**FIG. 16**



**FIG. 17**



**FIG. 18**

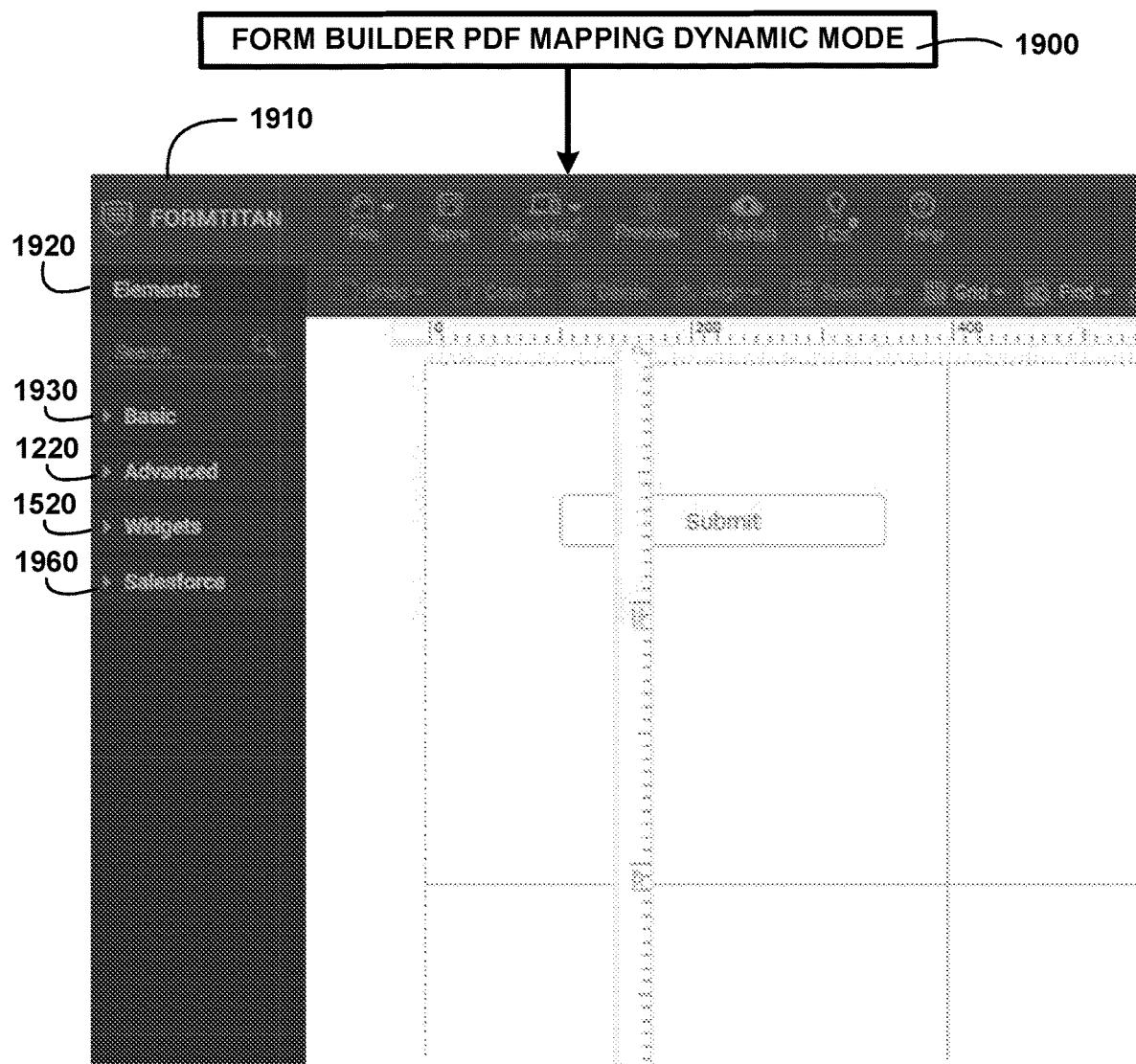
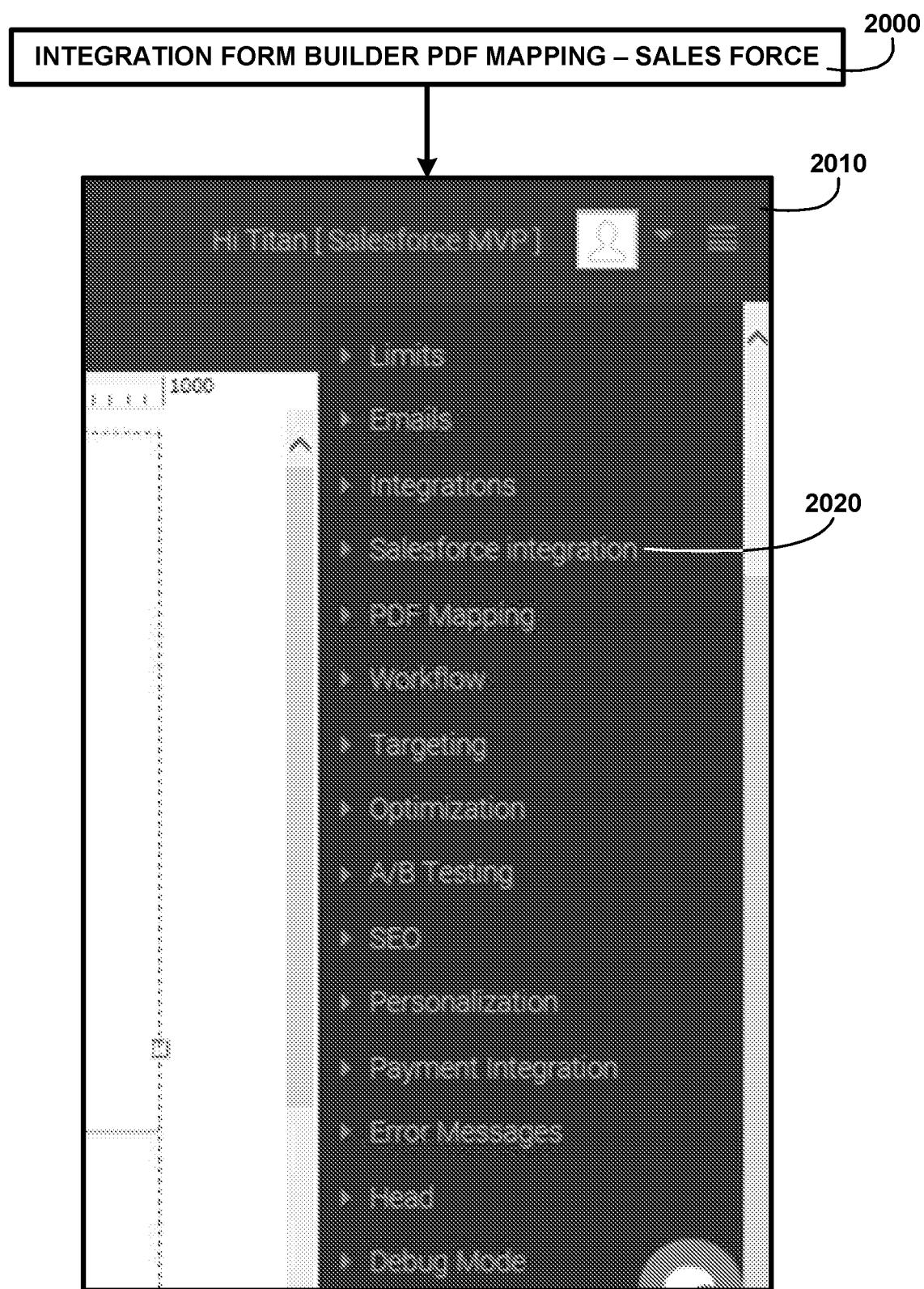
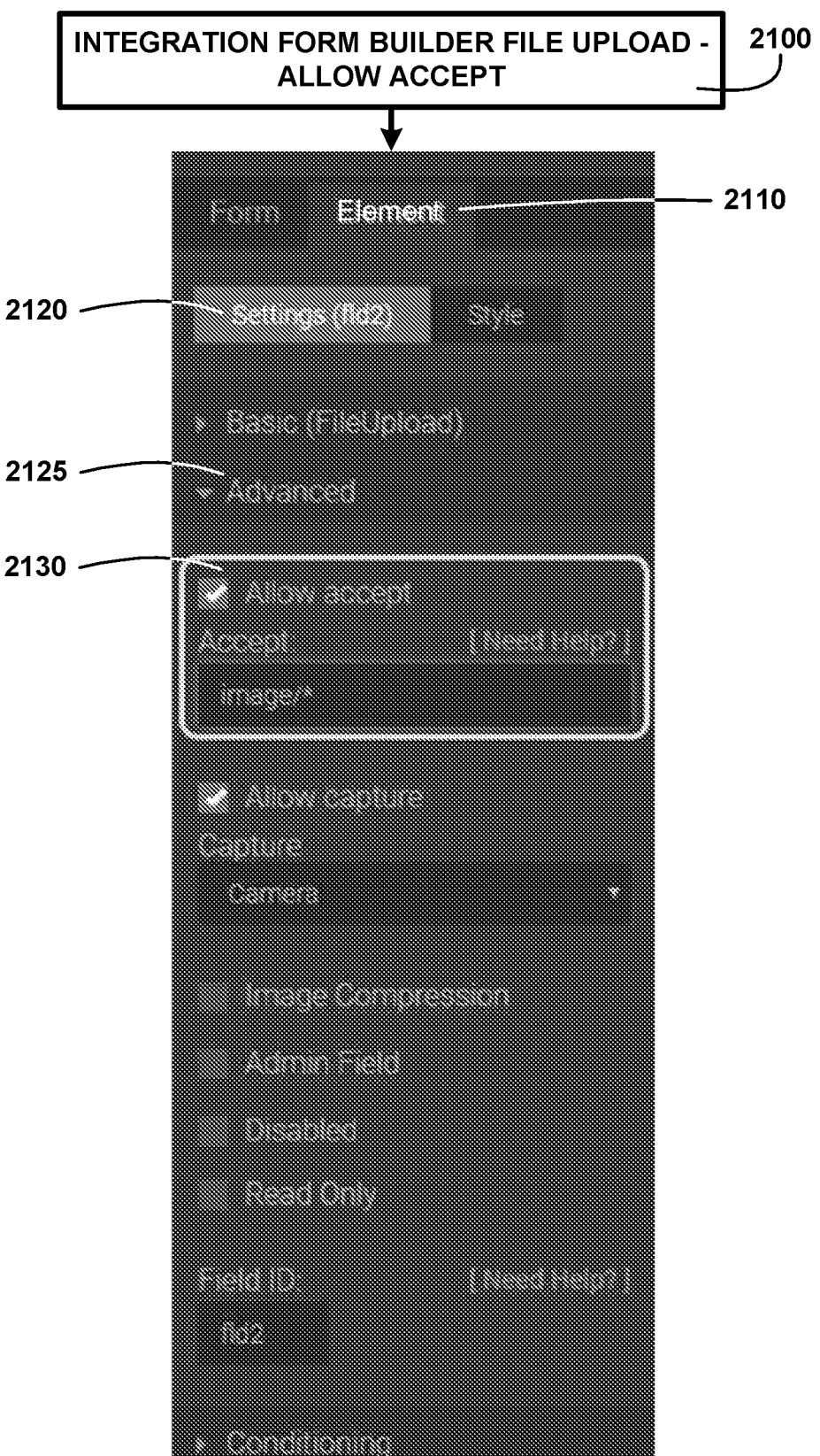


FIG. 19



**FIG. 20**



**FIG. 21**

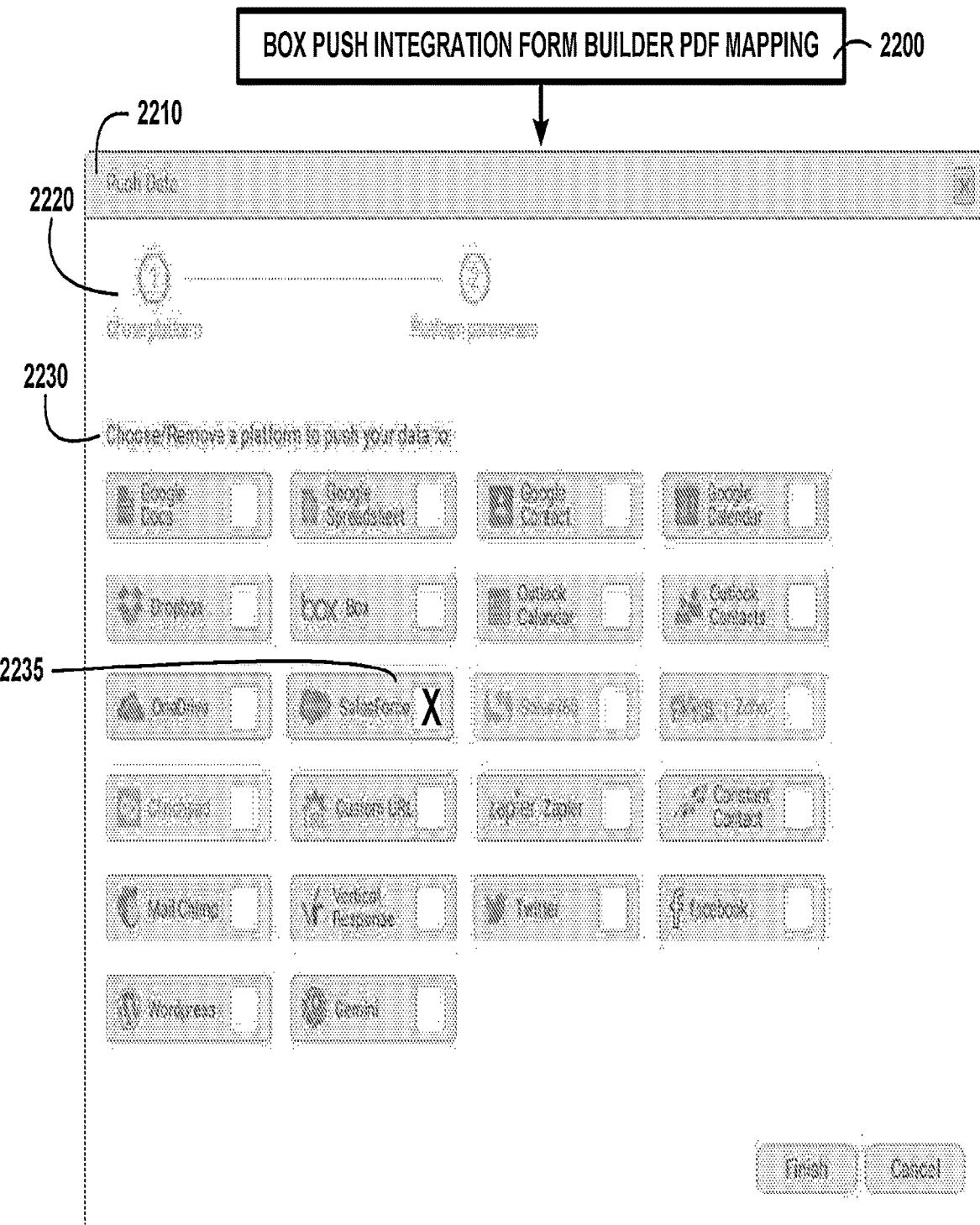


FIG. 22

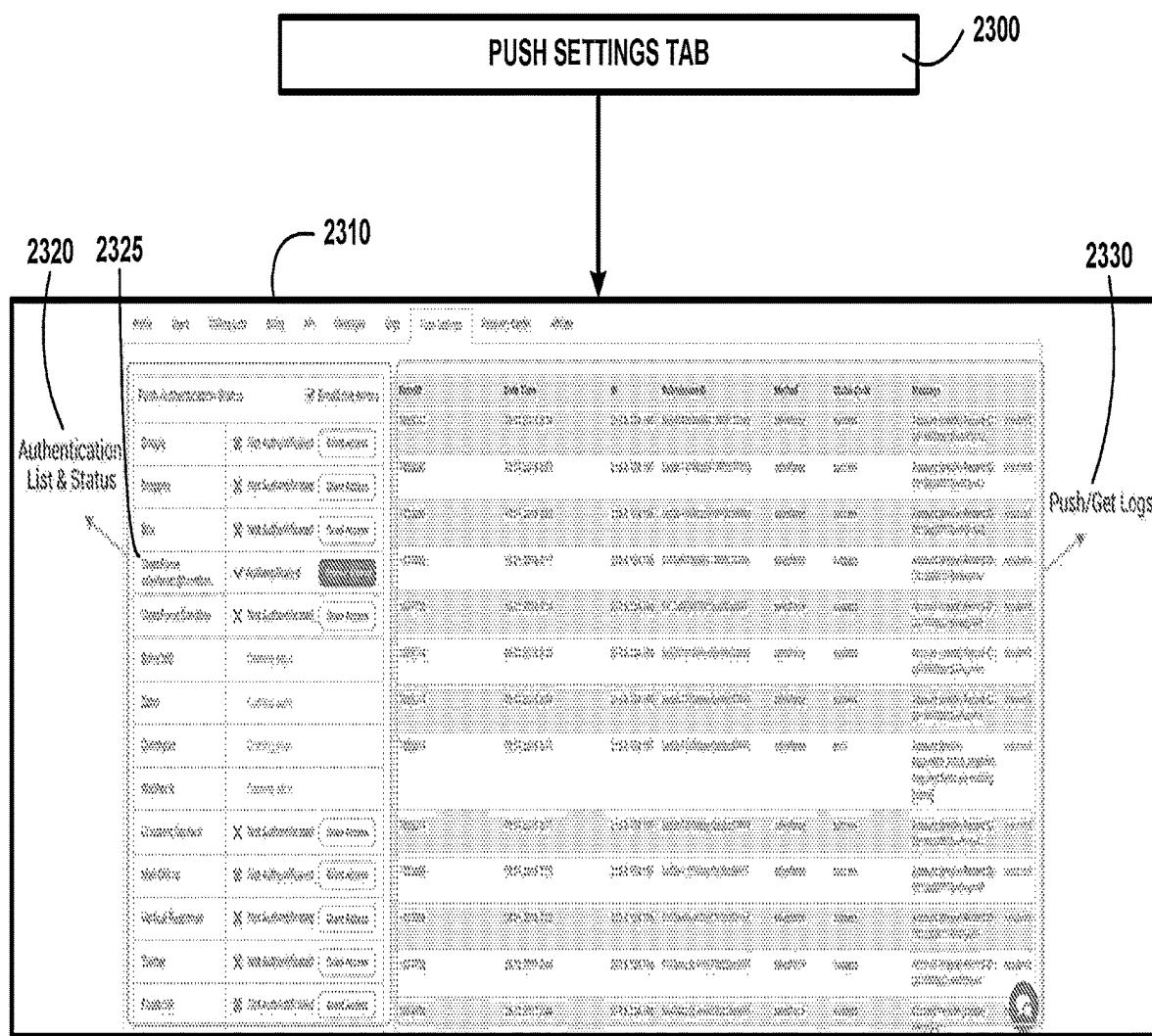
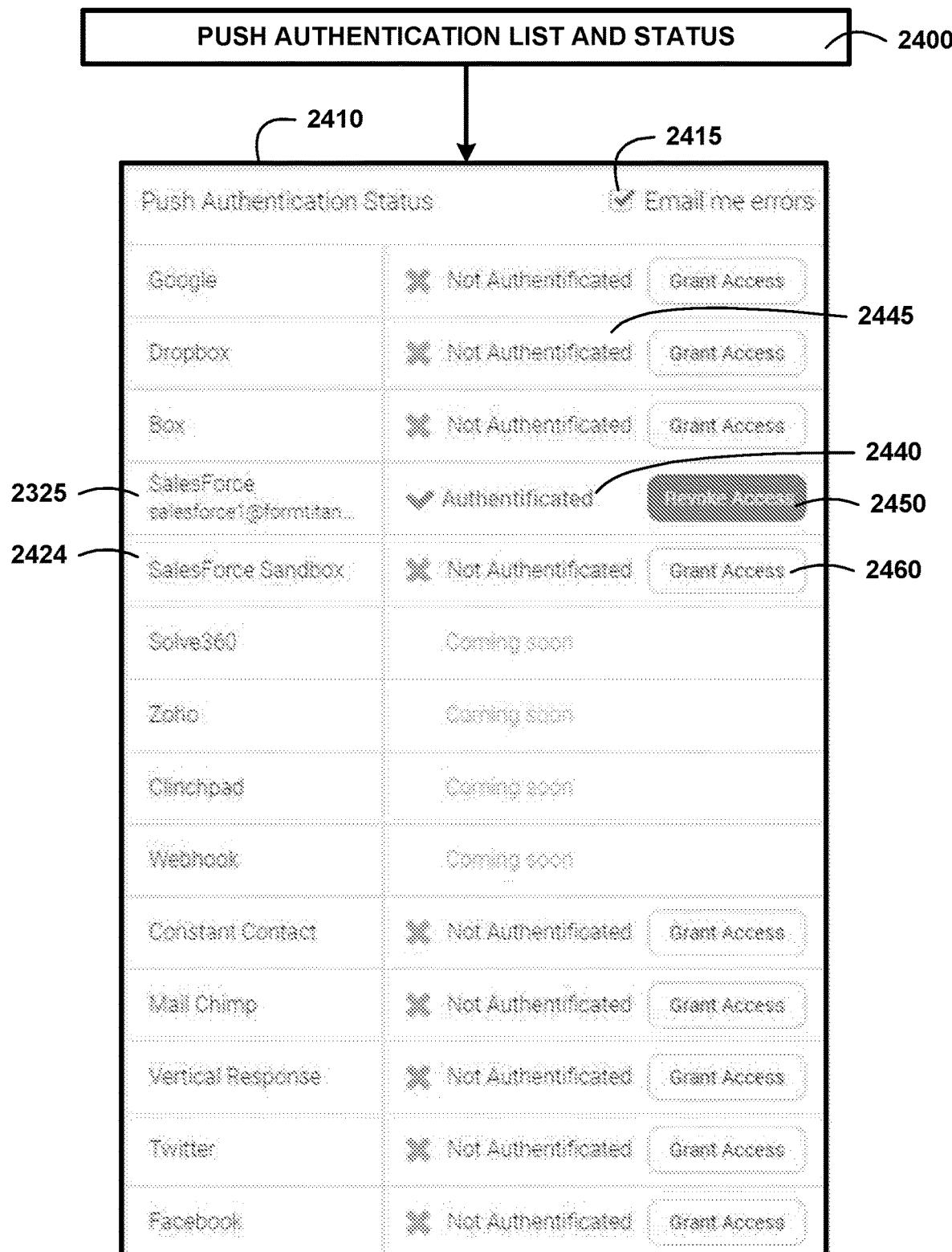


FIG. 23

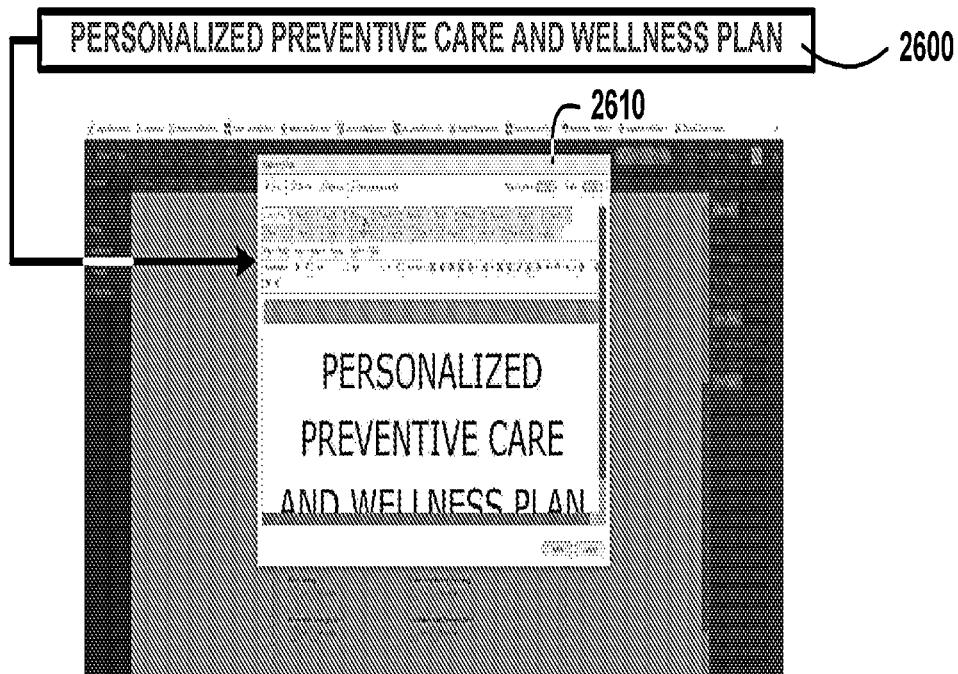


**FIG. 24**

**PUSH LOGS**

ID	Date/Time	IP	Submission ID	Method	Status Code	Message
12345	2022-08-04 10:00:00	192.168.1.1	12345678901234567890	POST	200	Success - Data processed.
12346	2022-08-04 10:01:00	192.168.1.2	12345678901234567891	POST	200	Success - Data processed.
12347	2022-08-04 10:02:00	192.168.1.3	12345678901234567892	POST	200	Success - Data processed.
12348	2022-08-04 10:03:00	192.168.1.4	12345678901234567893	POST	200	Success - Data processed.
12349	2022-08-04 10:04:00	192.168.1.5	12345678901234567894	POST	200	Success - Data processed.
12350	2022-08-04 10:05:00	192.168.1.6	12345678901234567895	POST	200	Success - Data processed.

**FIG. 25**



**FIG. 26**

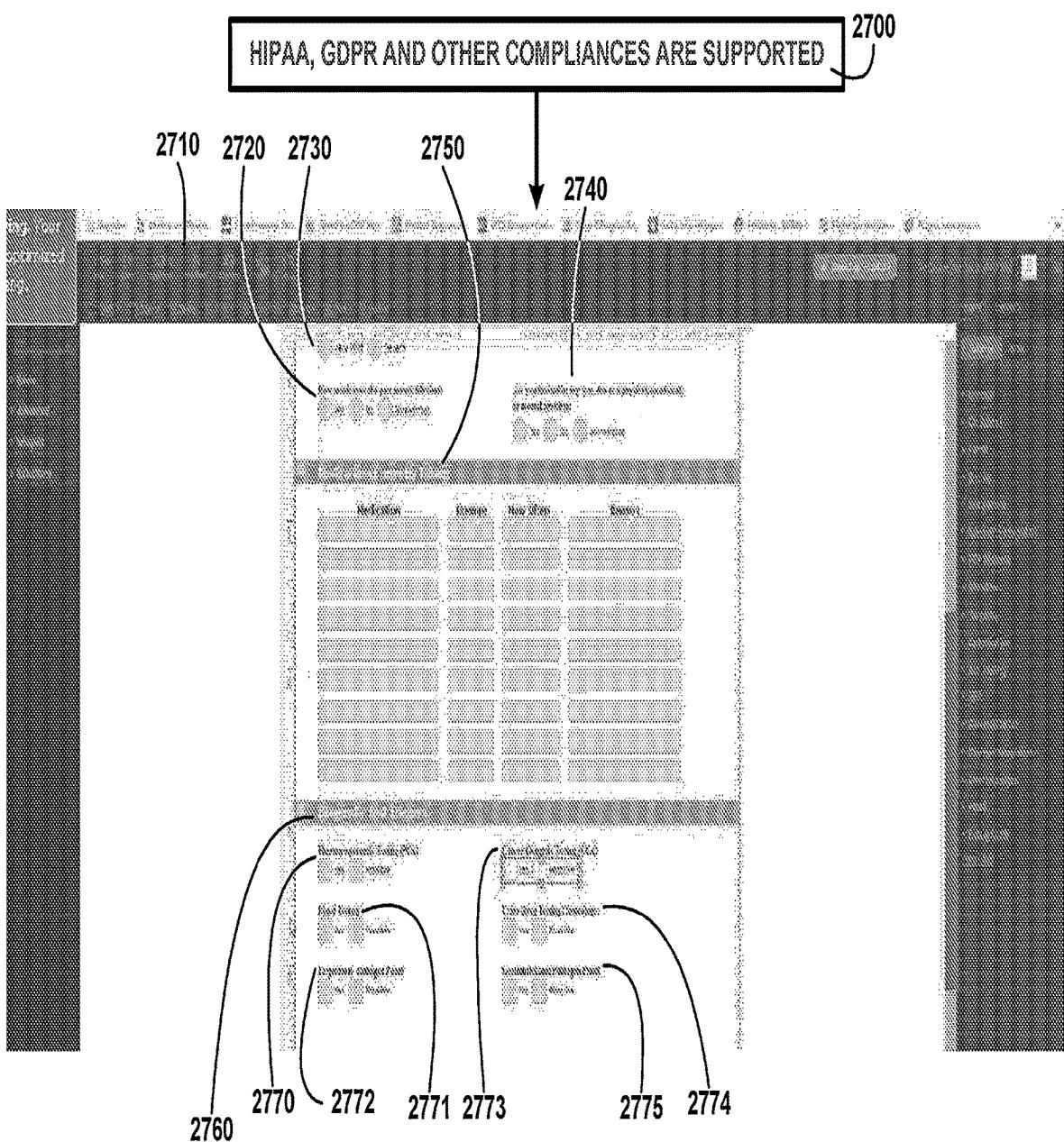


FIG. 27

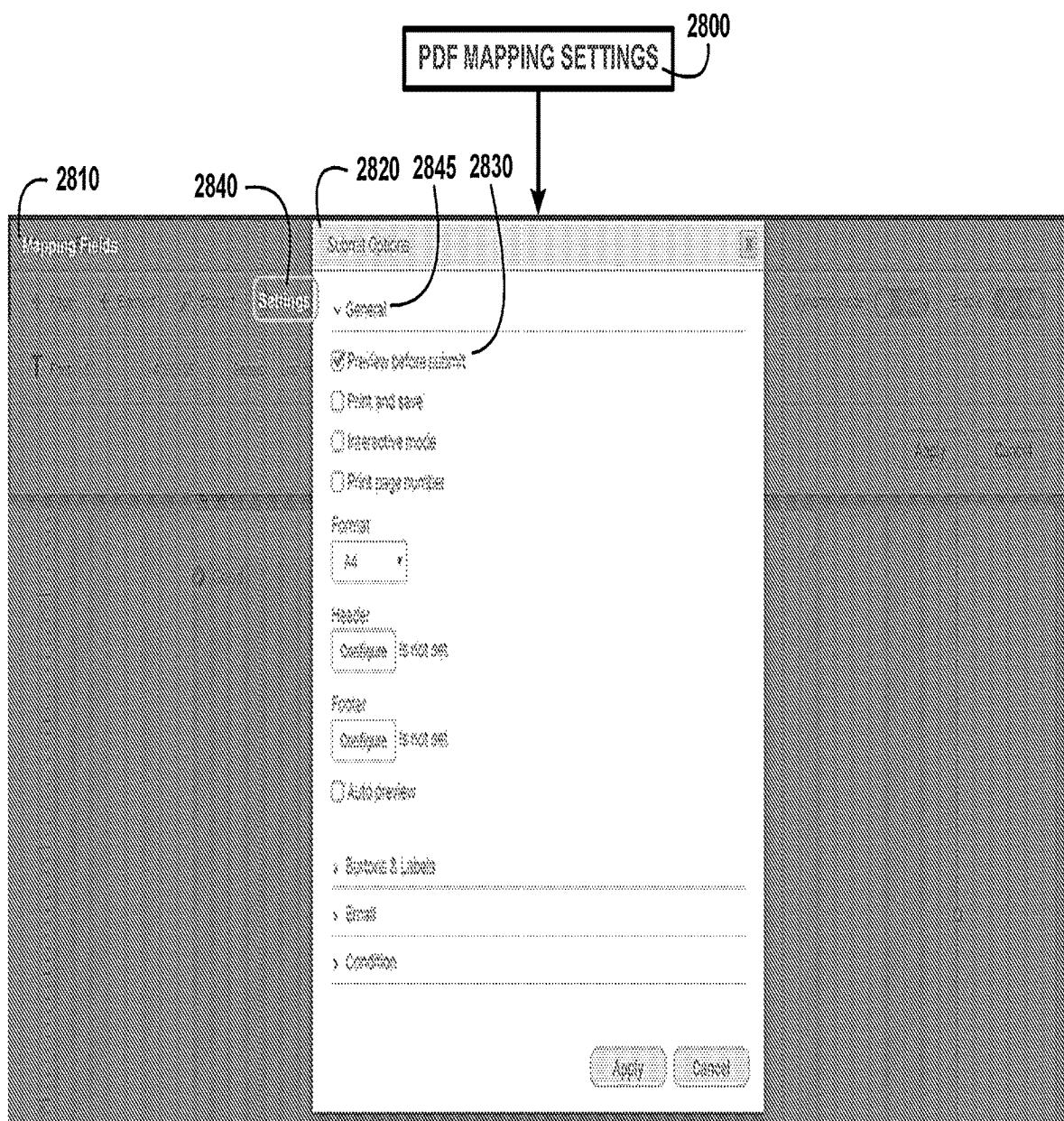
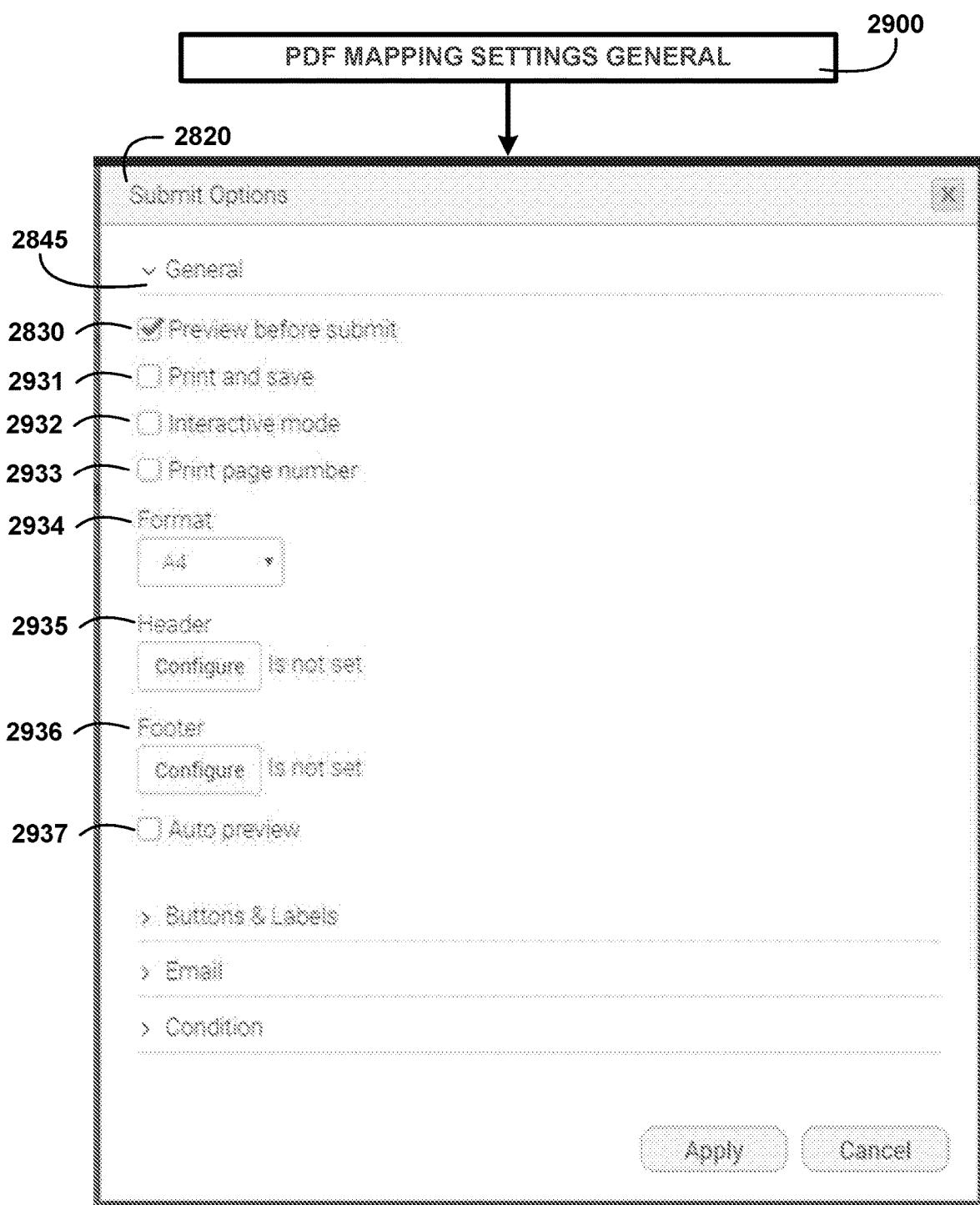


FIG. 28



**FIG. 29**

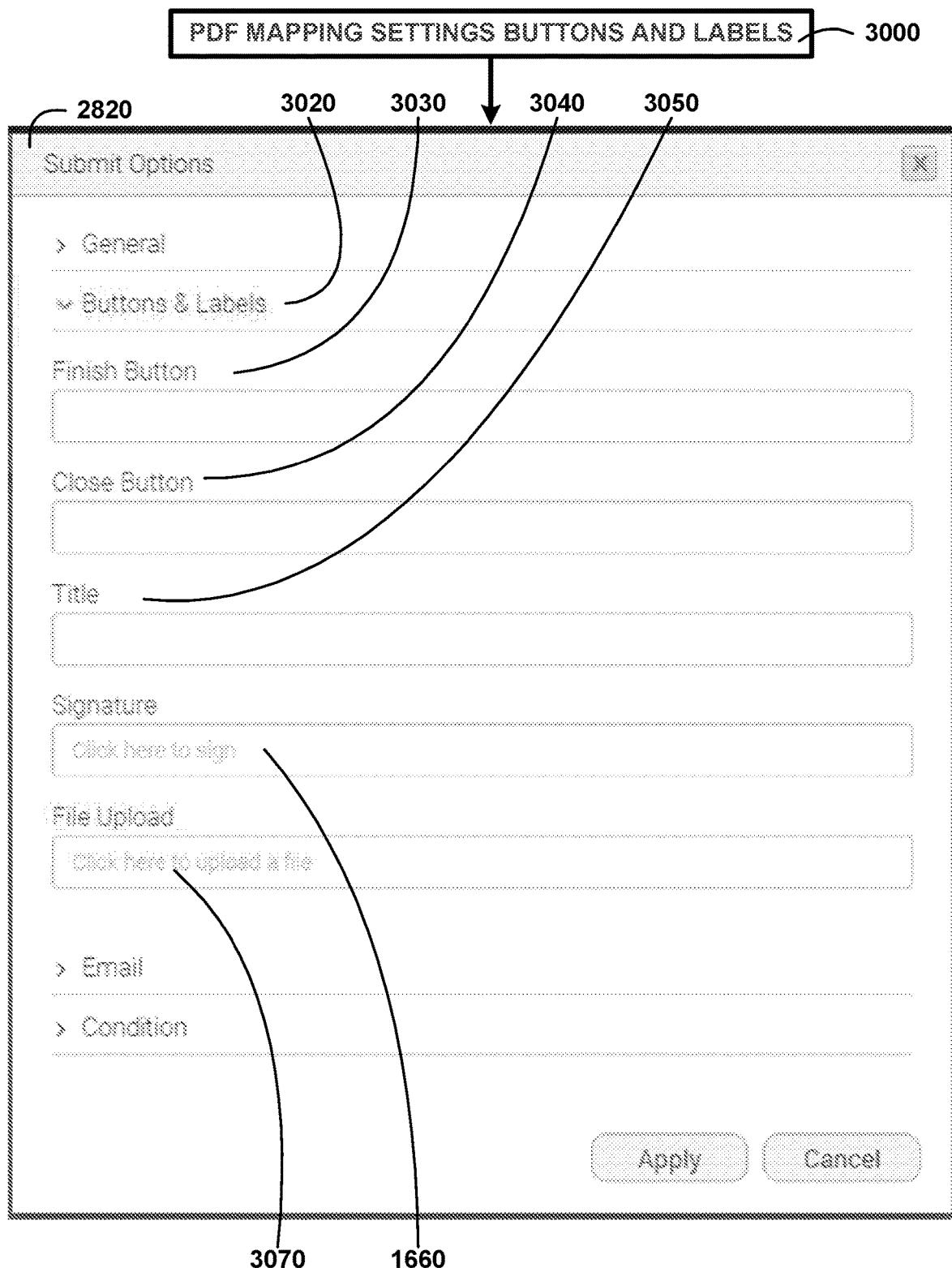
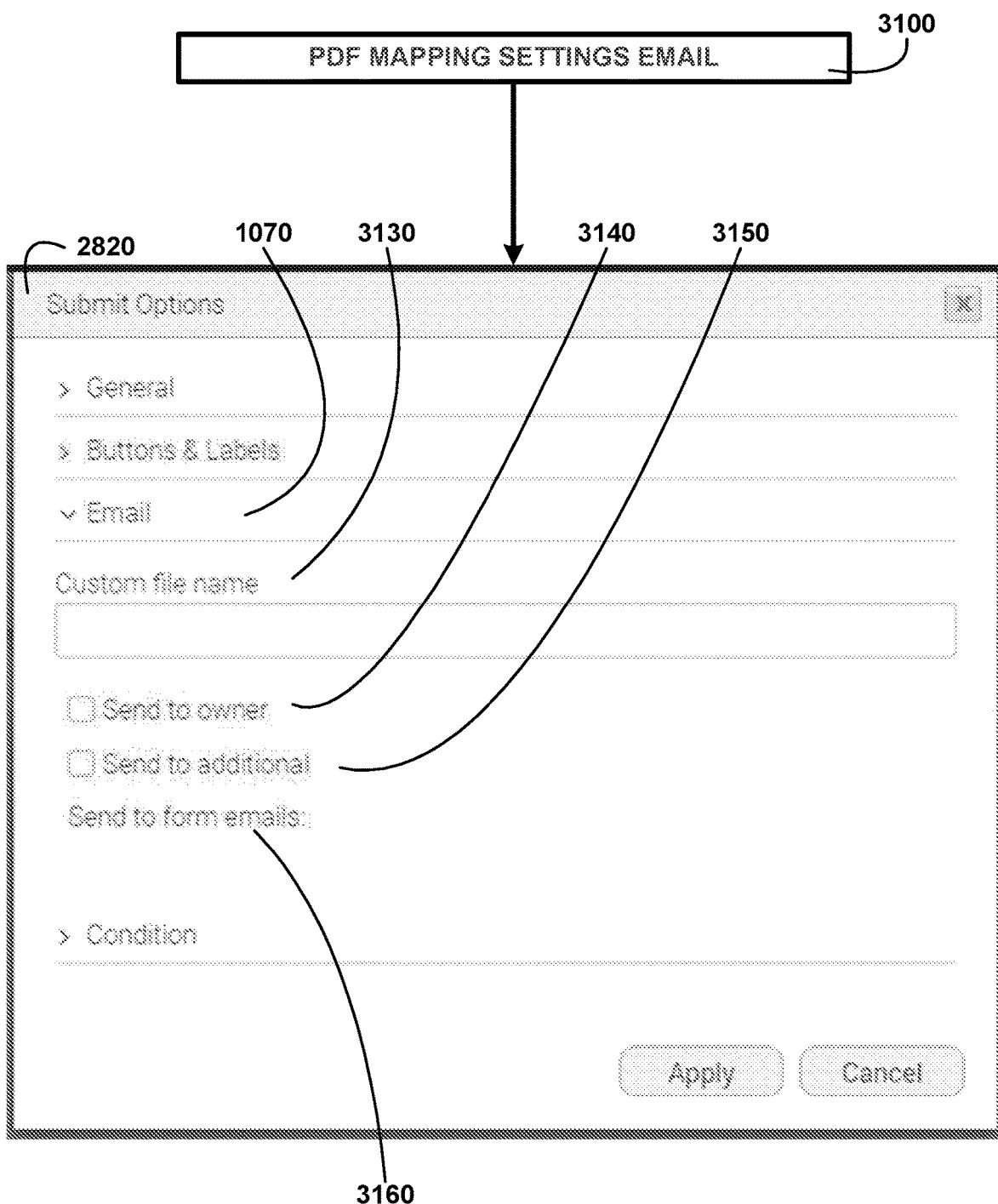
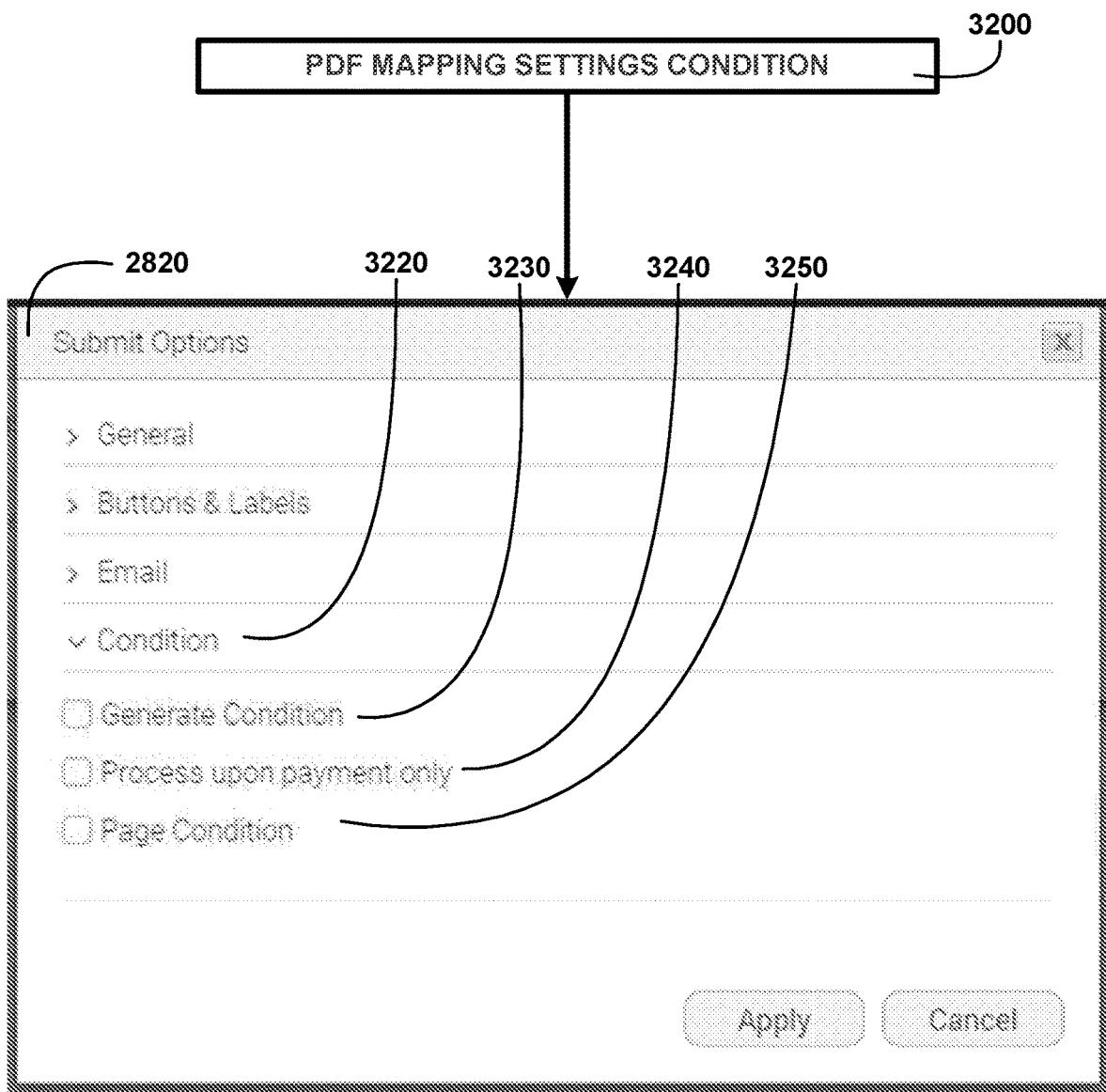


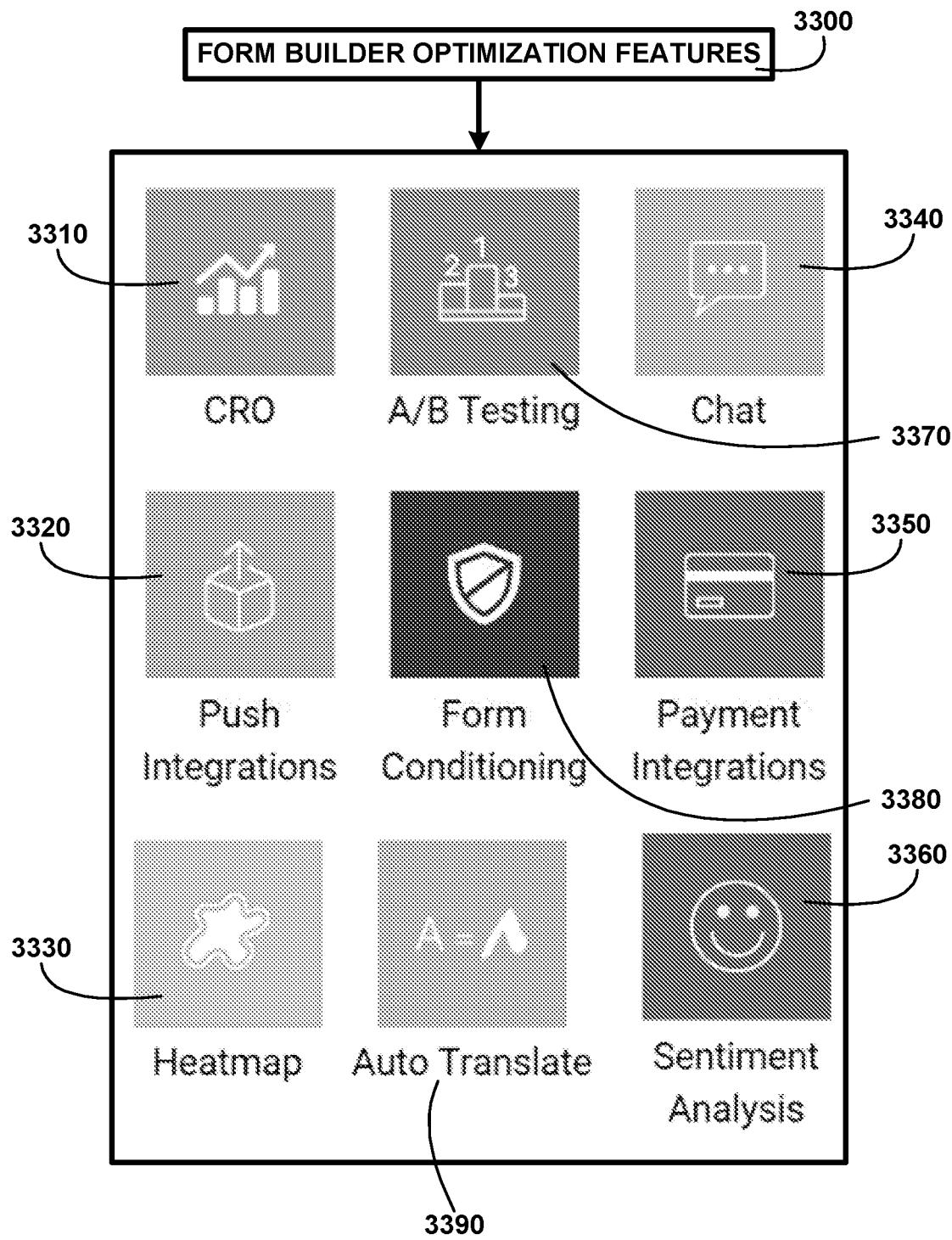
FIG. 30



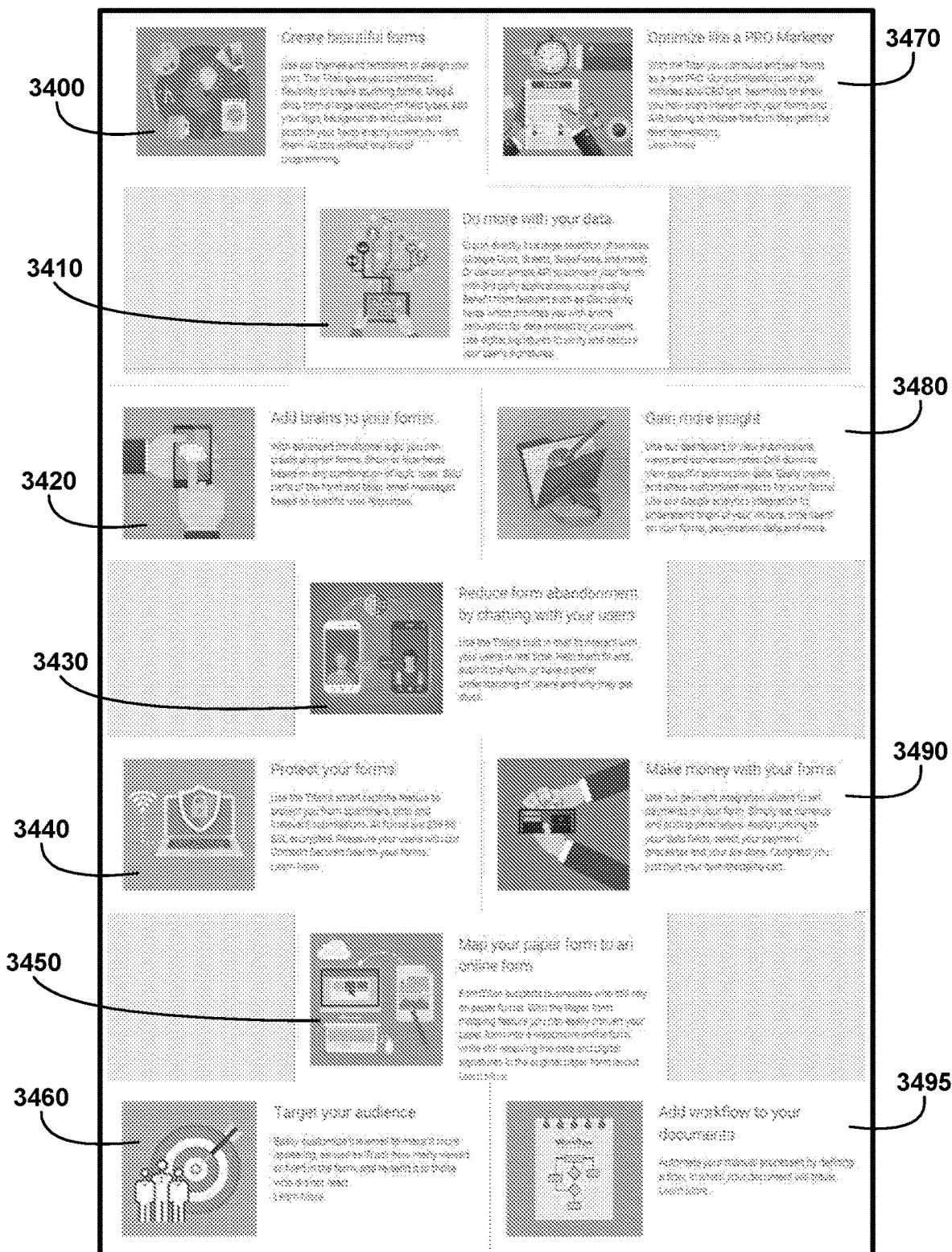
**FIG. 31**



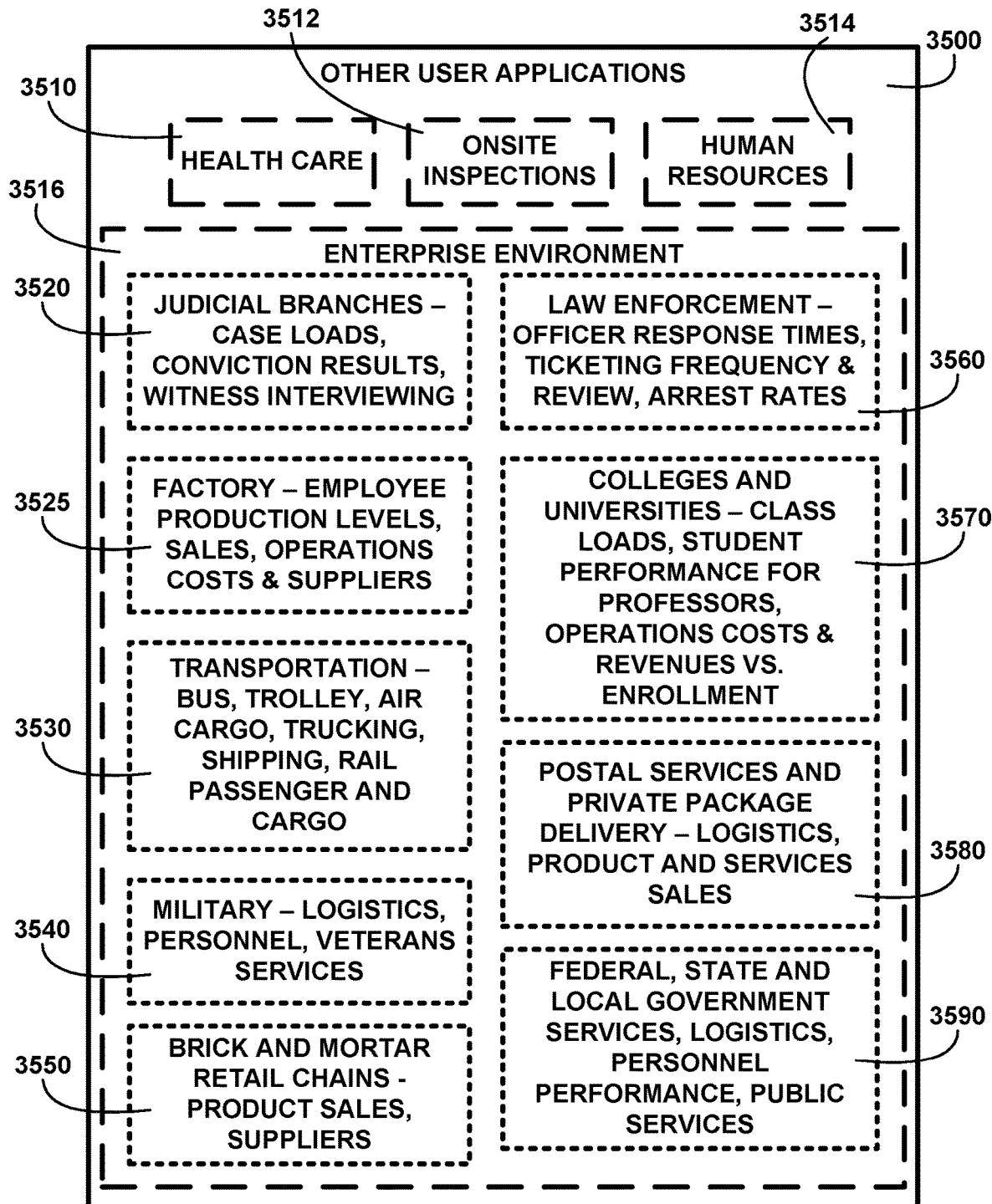
**FIG. 32**



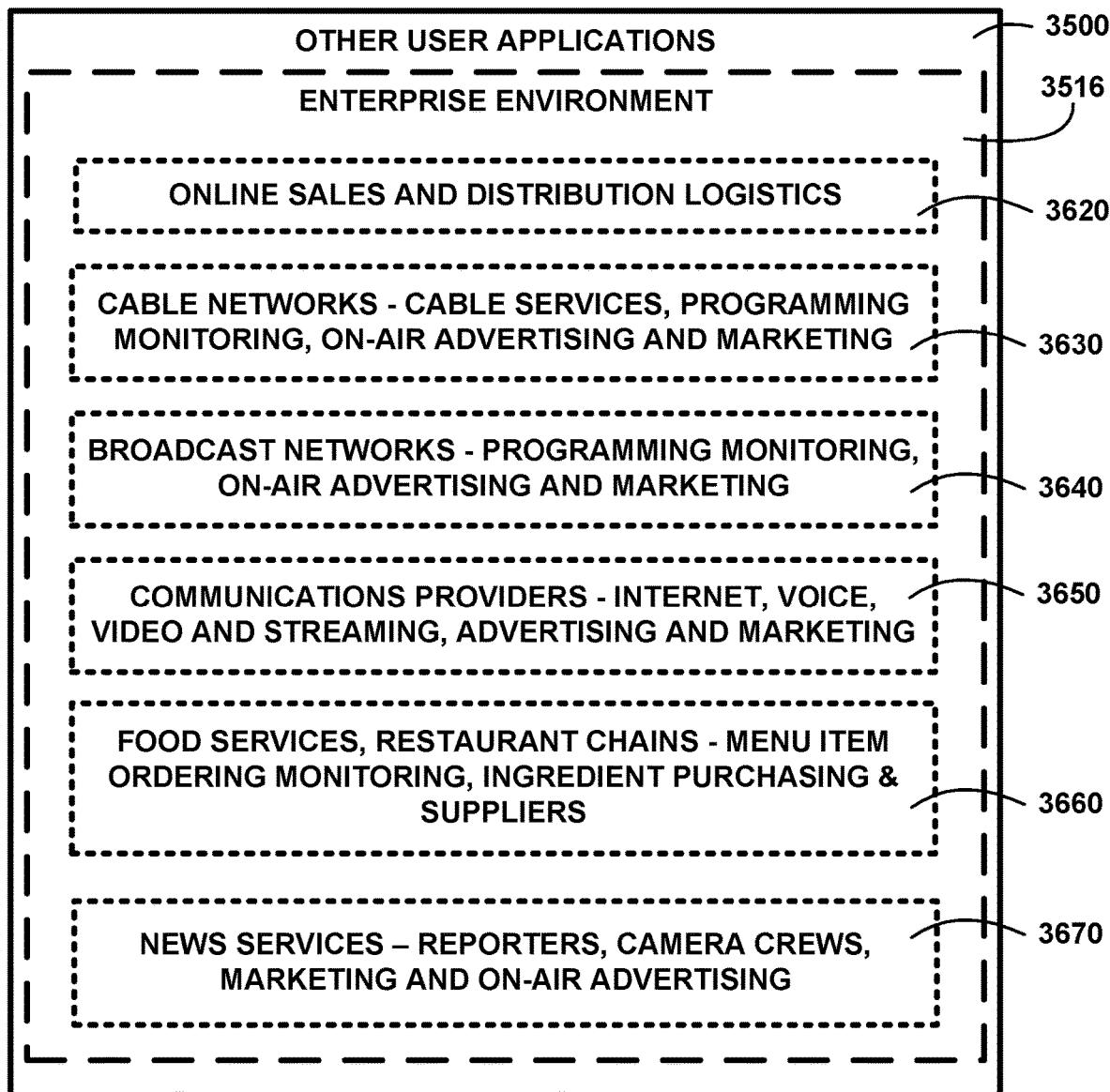
**FIG. 33**



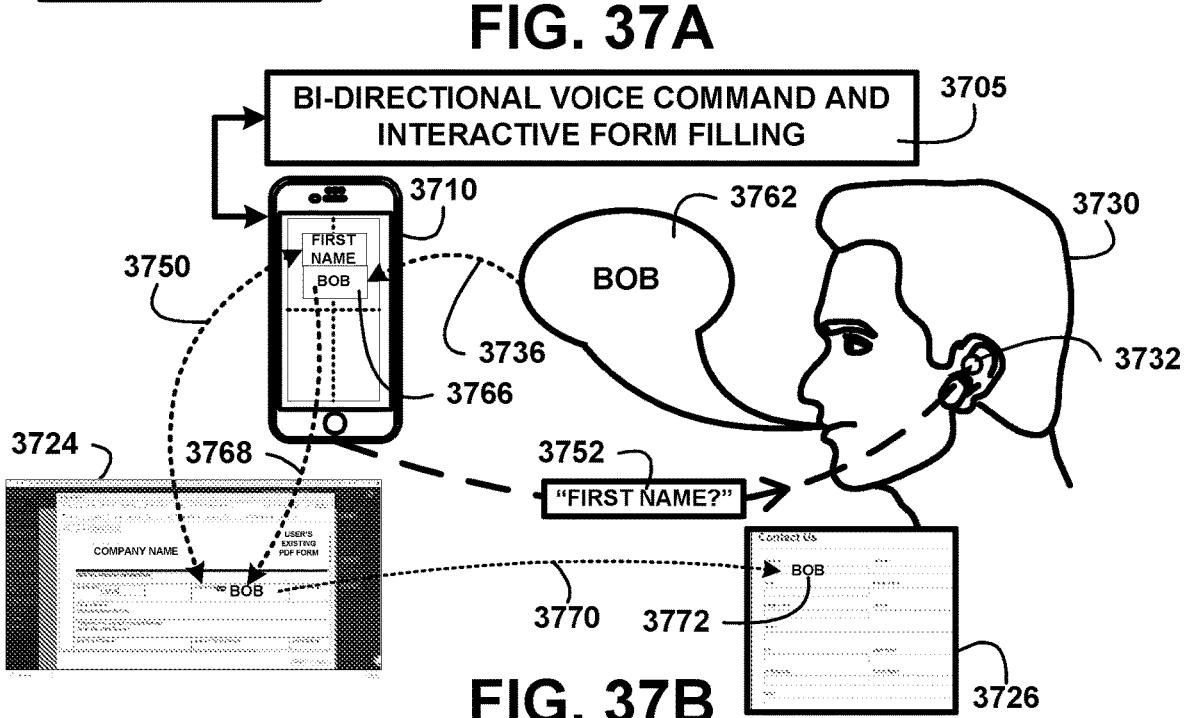
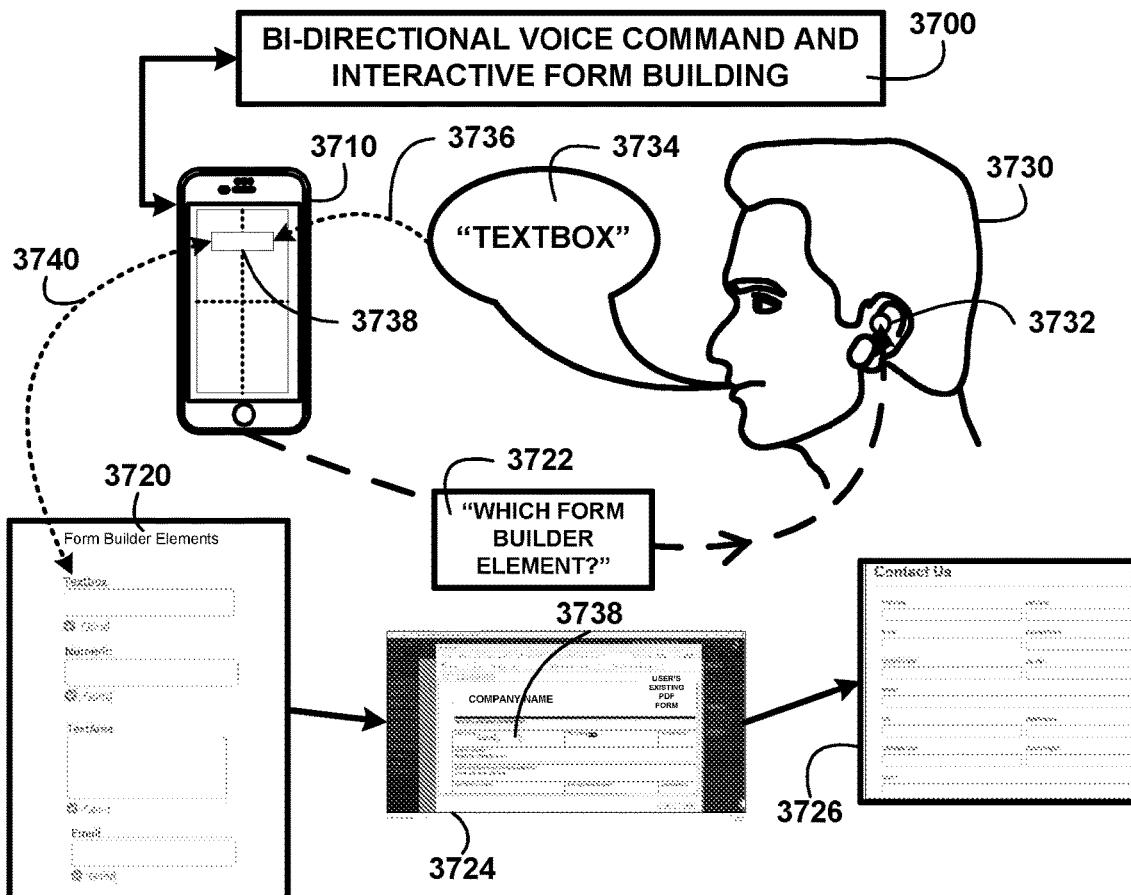
**FIG. 34**



**FIG. 35**



**FIG. 36**



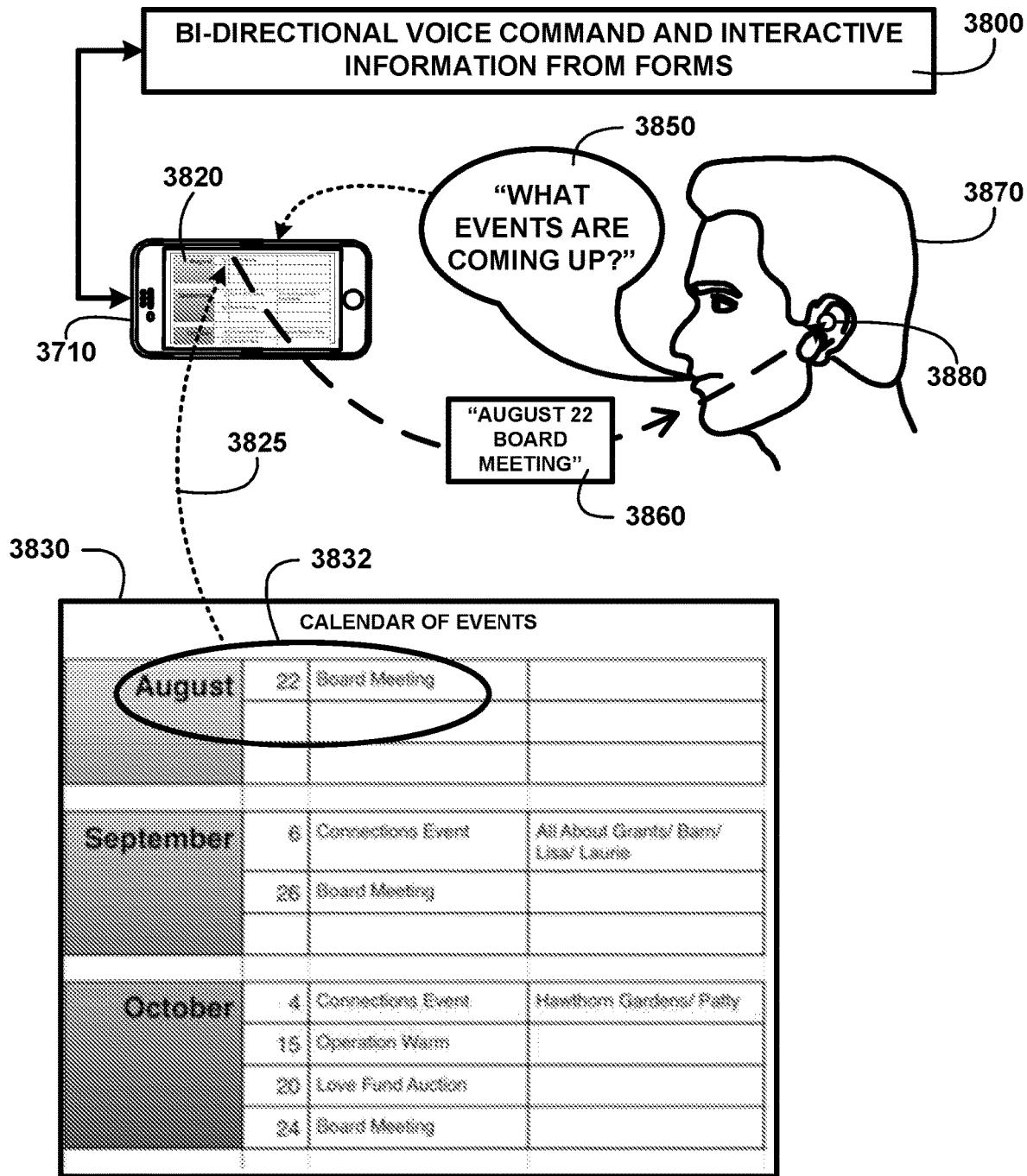
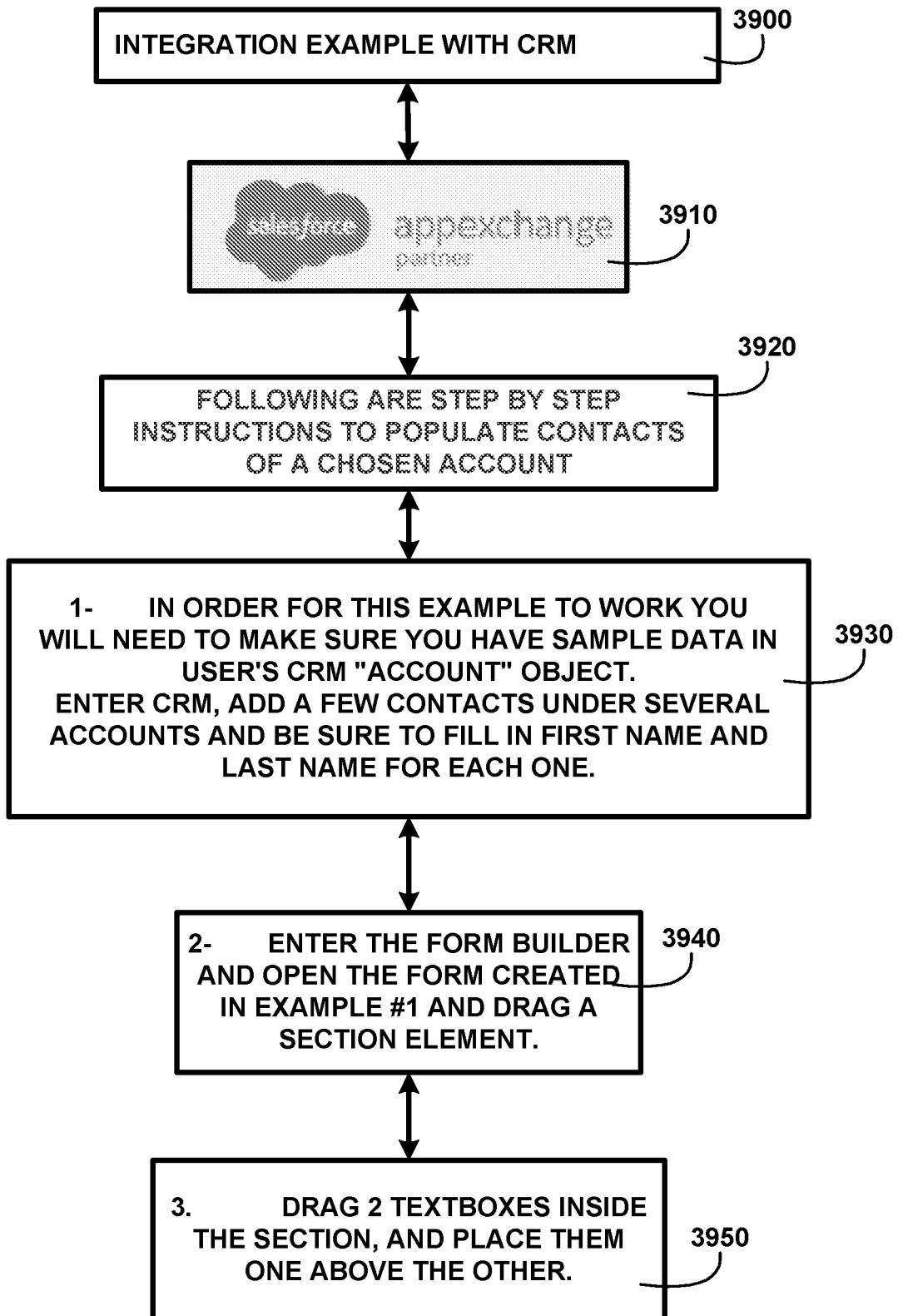
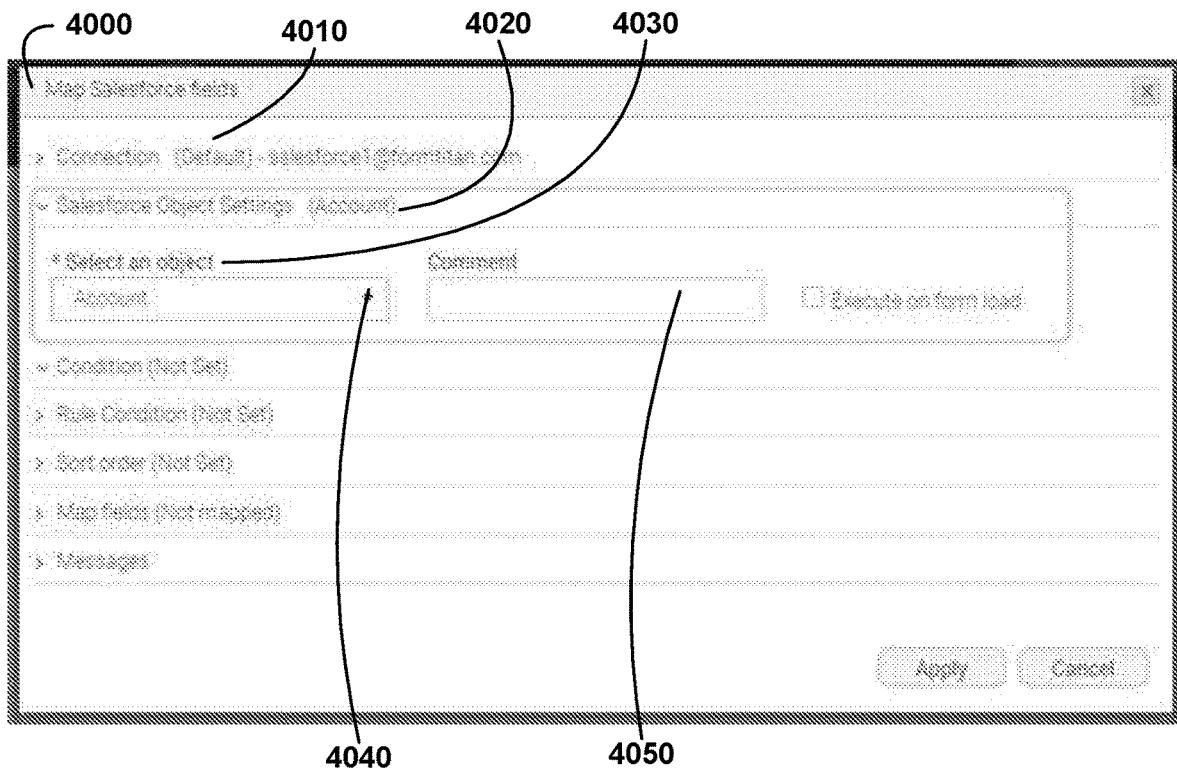


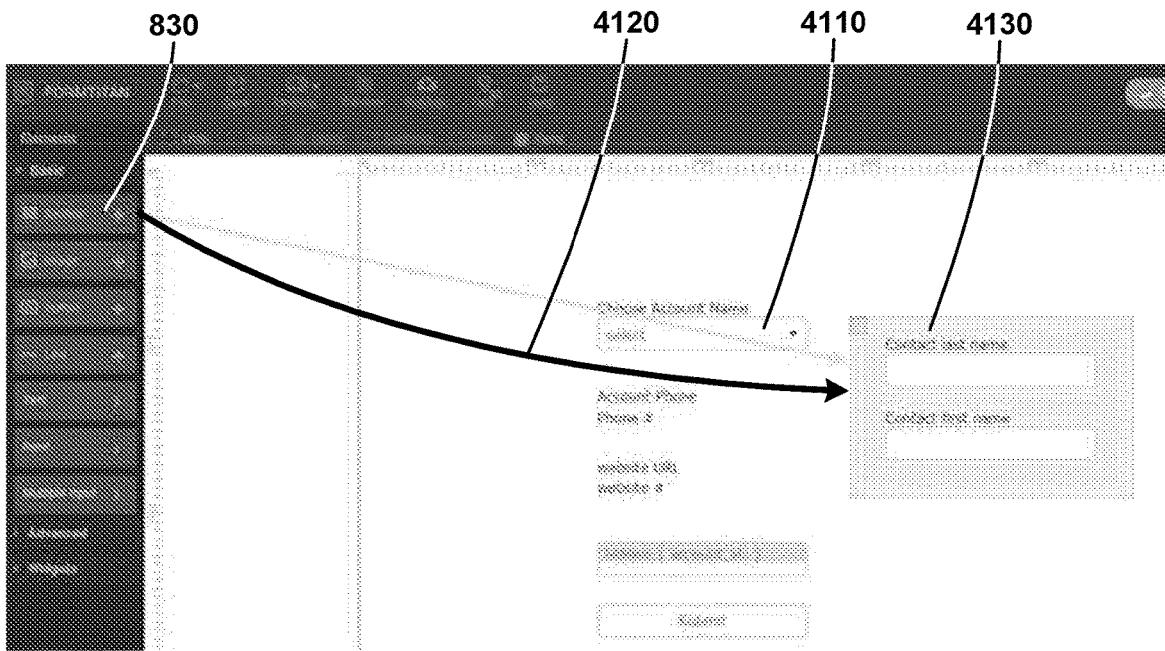
FIG. 38



**FIG. 39**



**FIG. 40**



**FIG. 41**

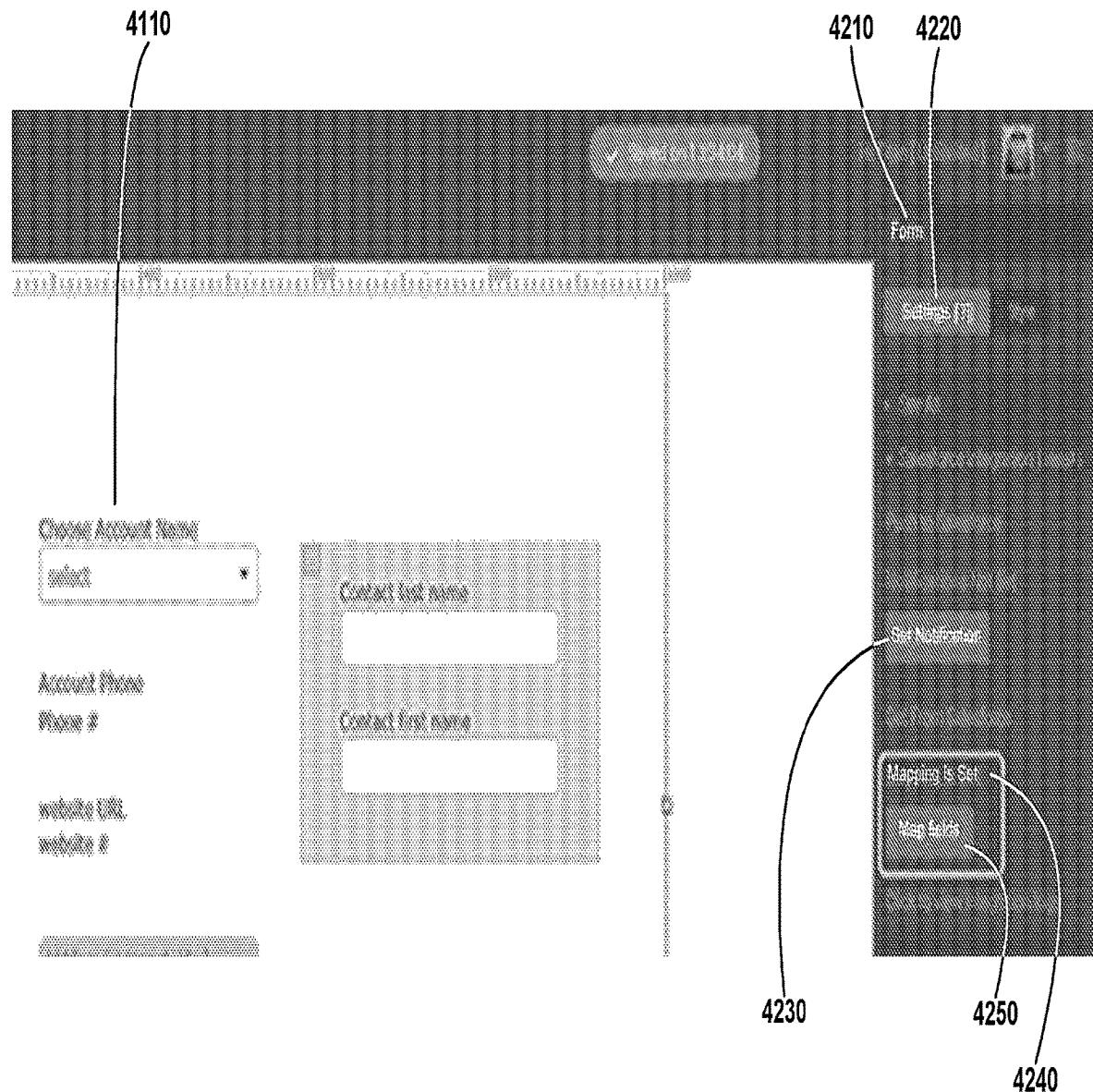


FIG. 42

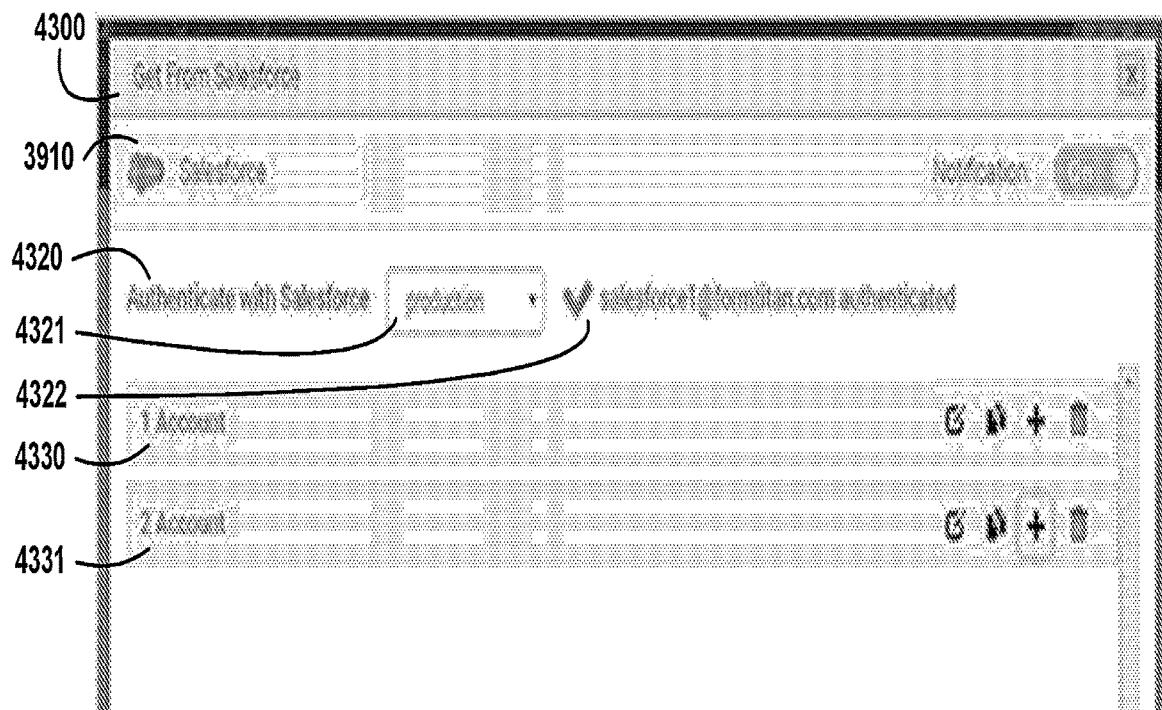


FIG. 43A

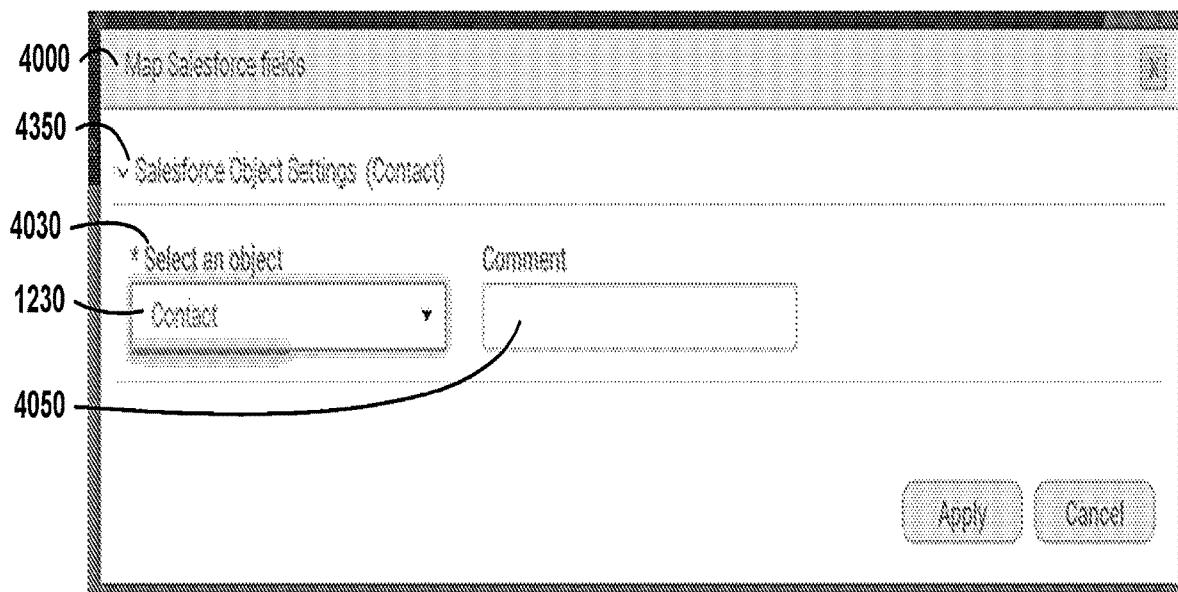


FIG. 43B

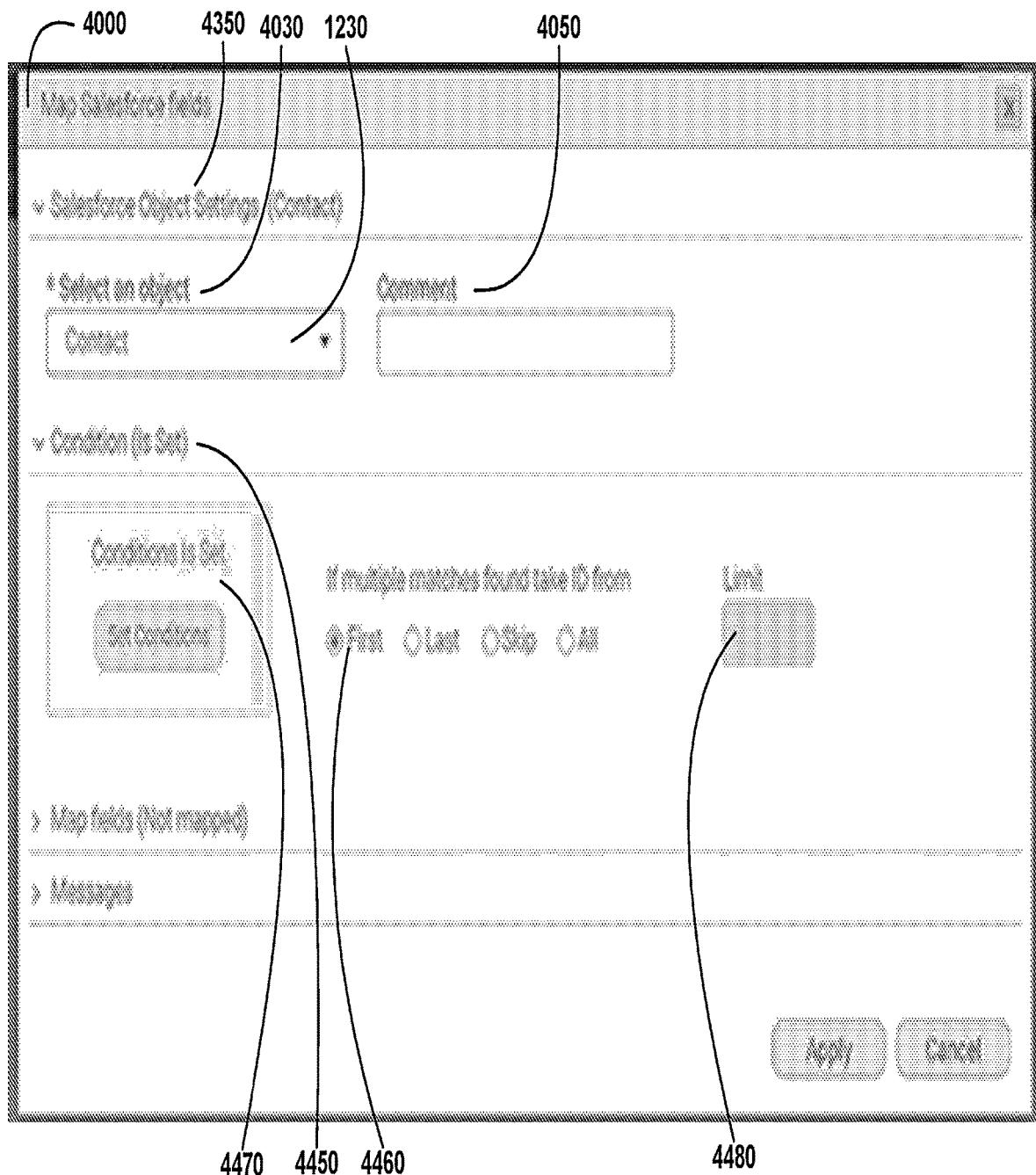


FIG. 44

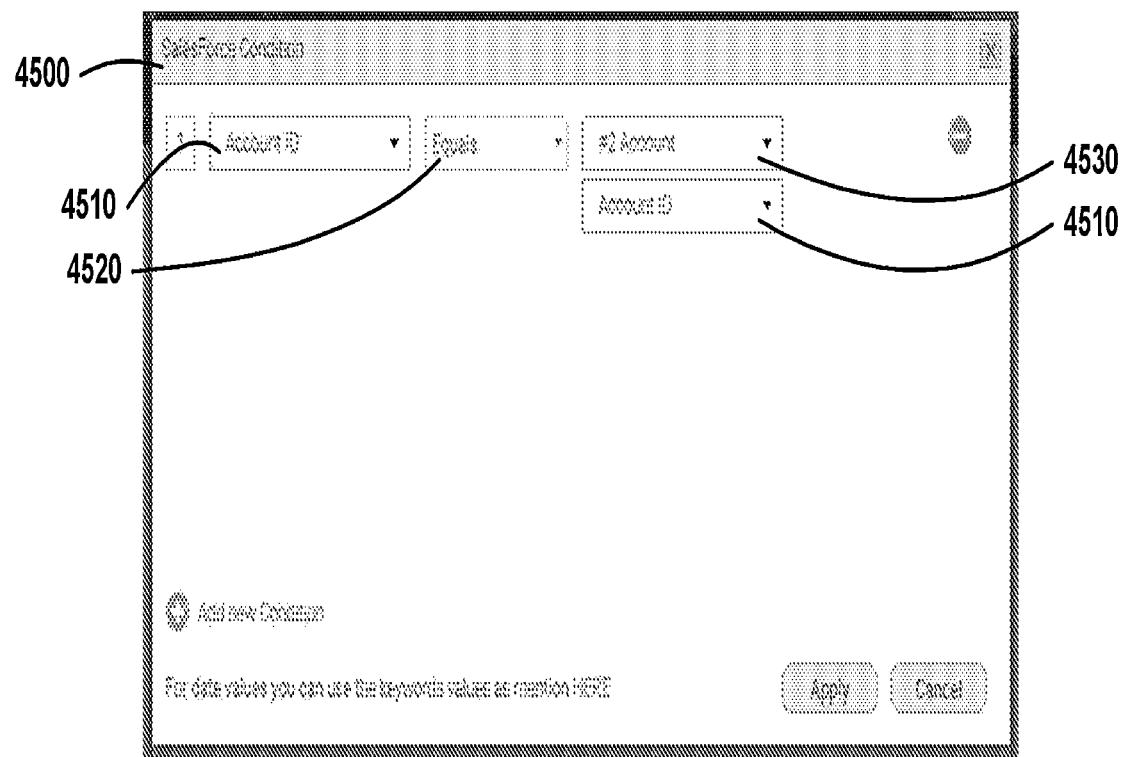


FIG. 45

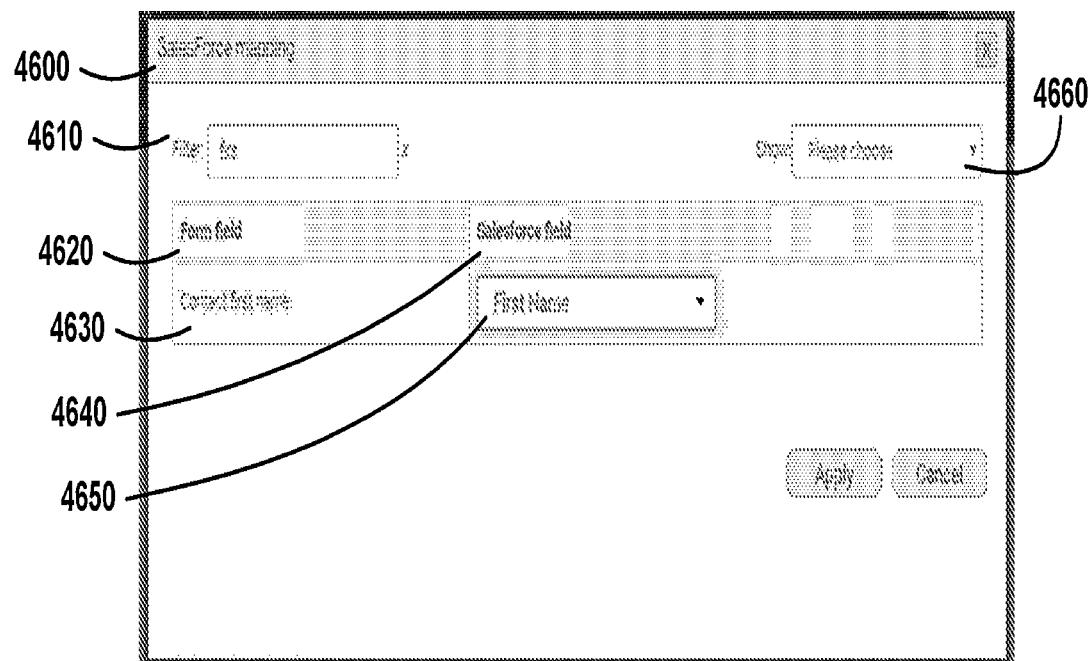


FIG. 46

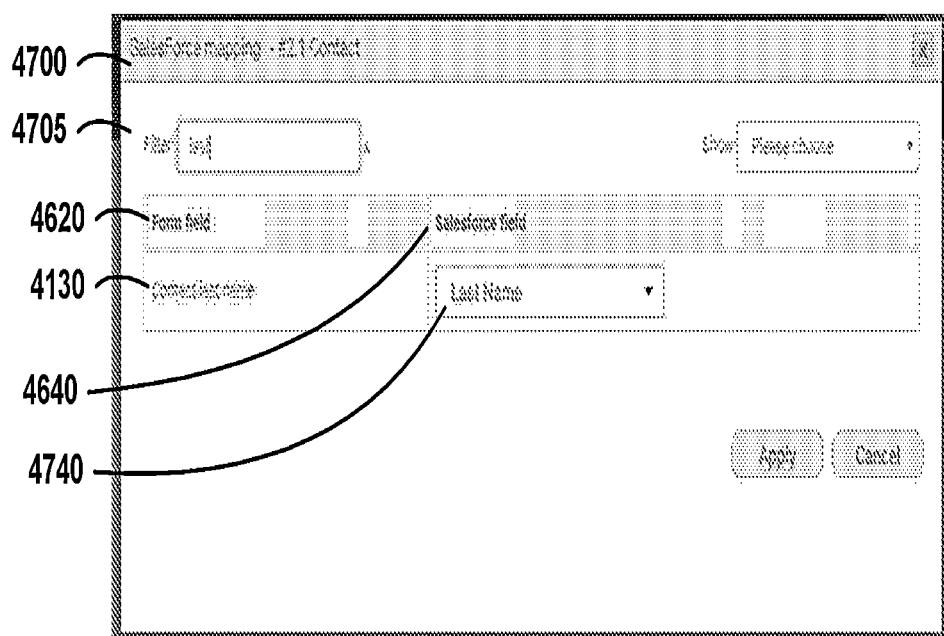


FIG. 47

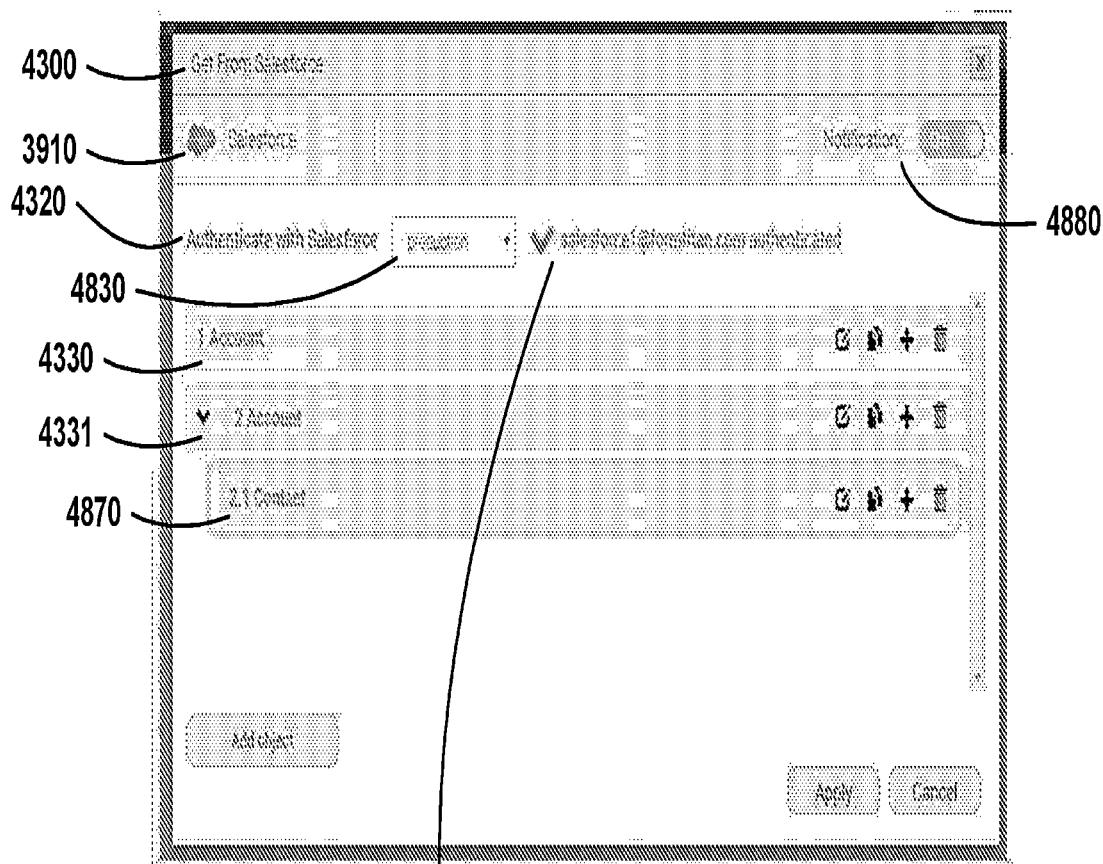
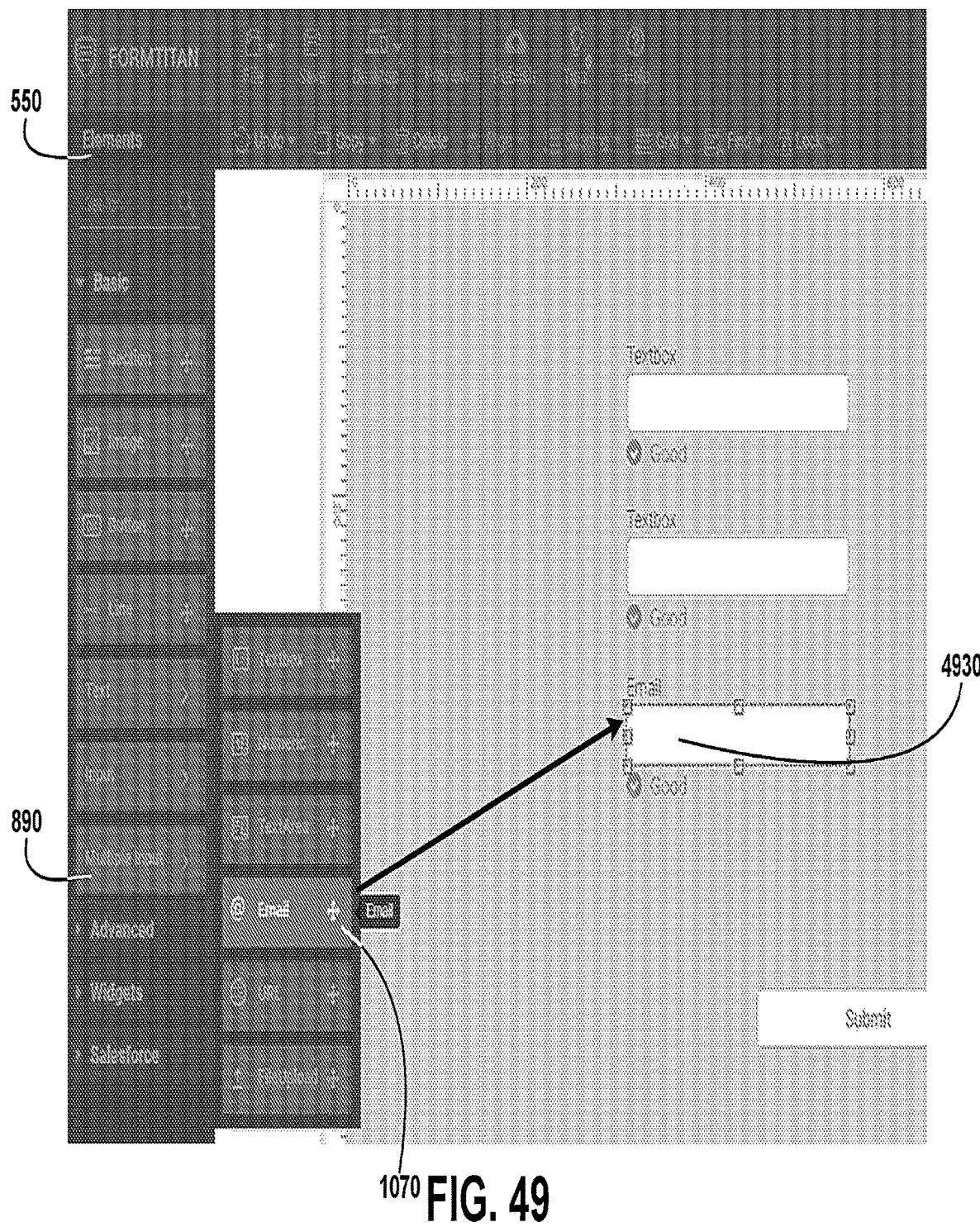


FIG. 48



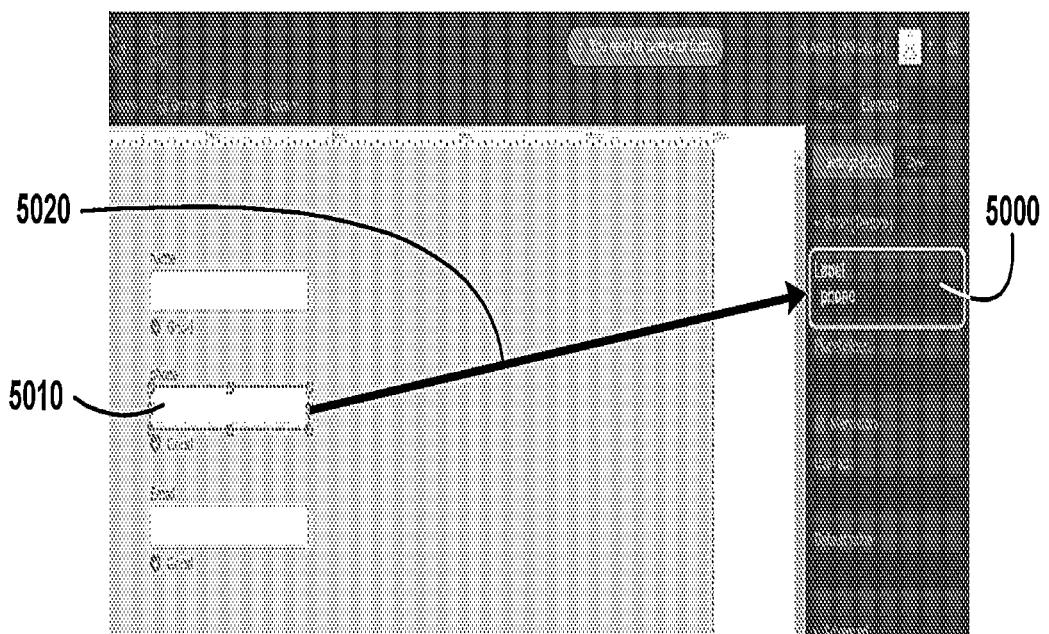


FIG. 50

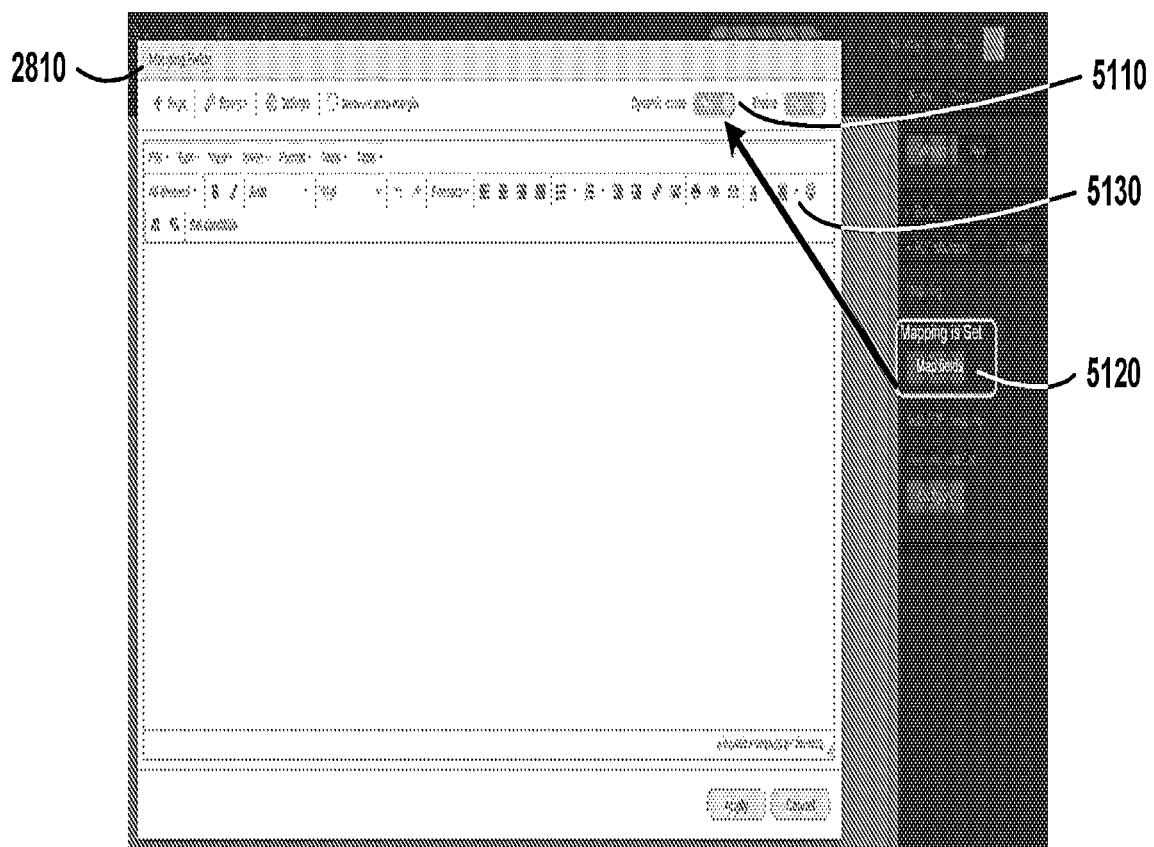


FIG. 51

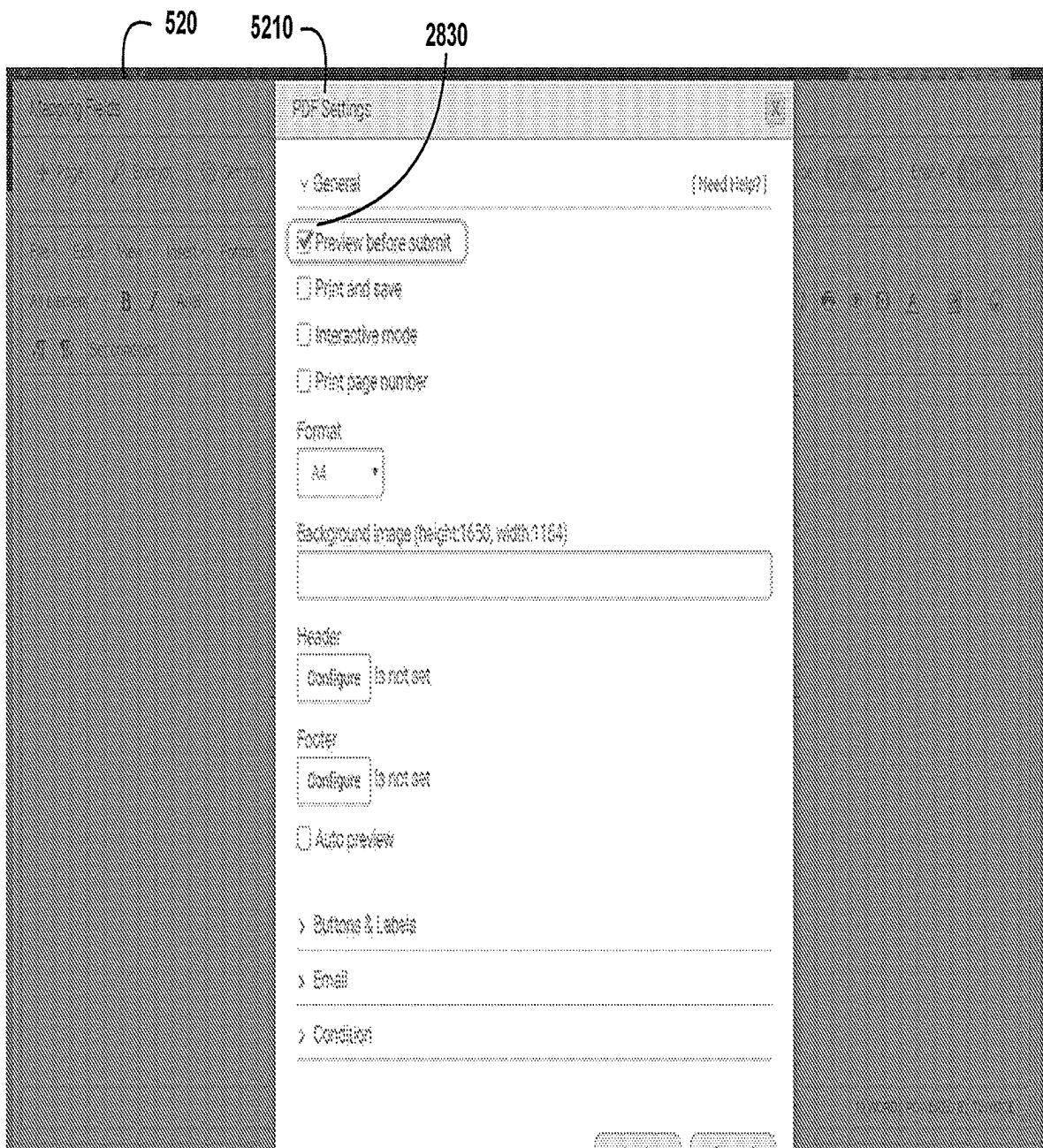
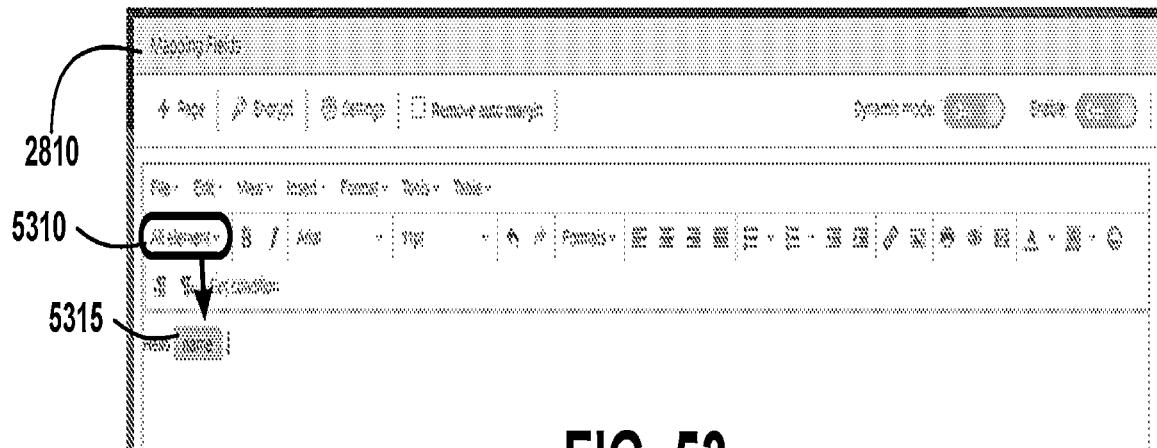
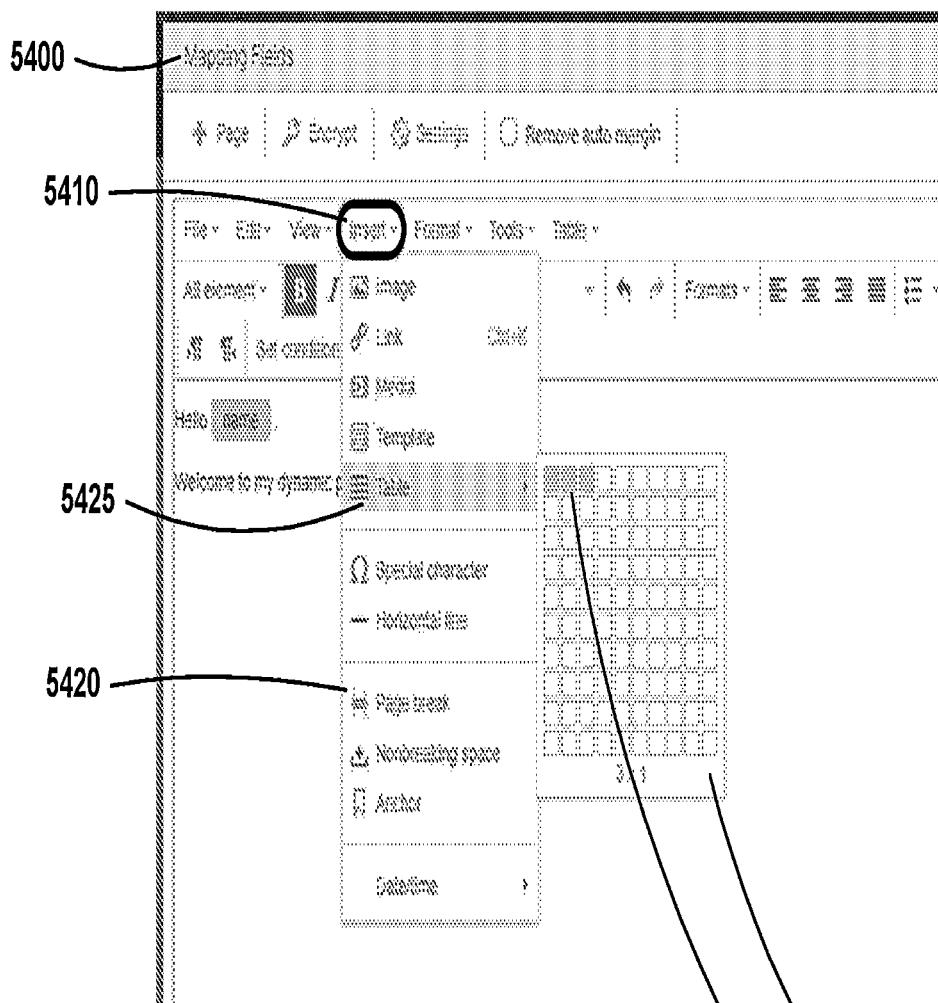


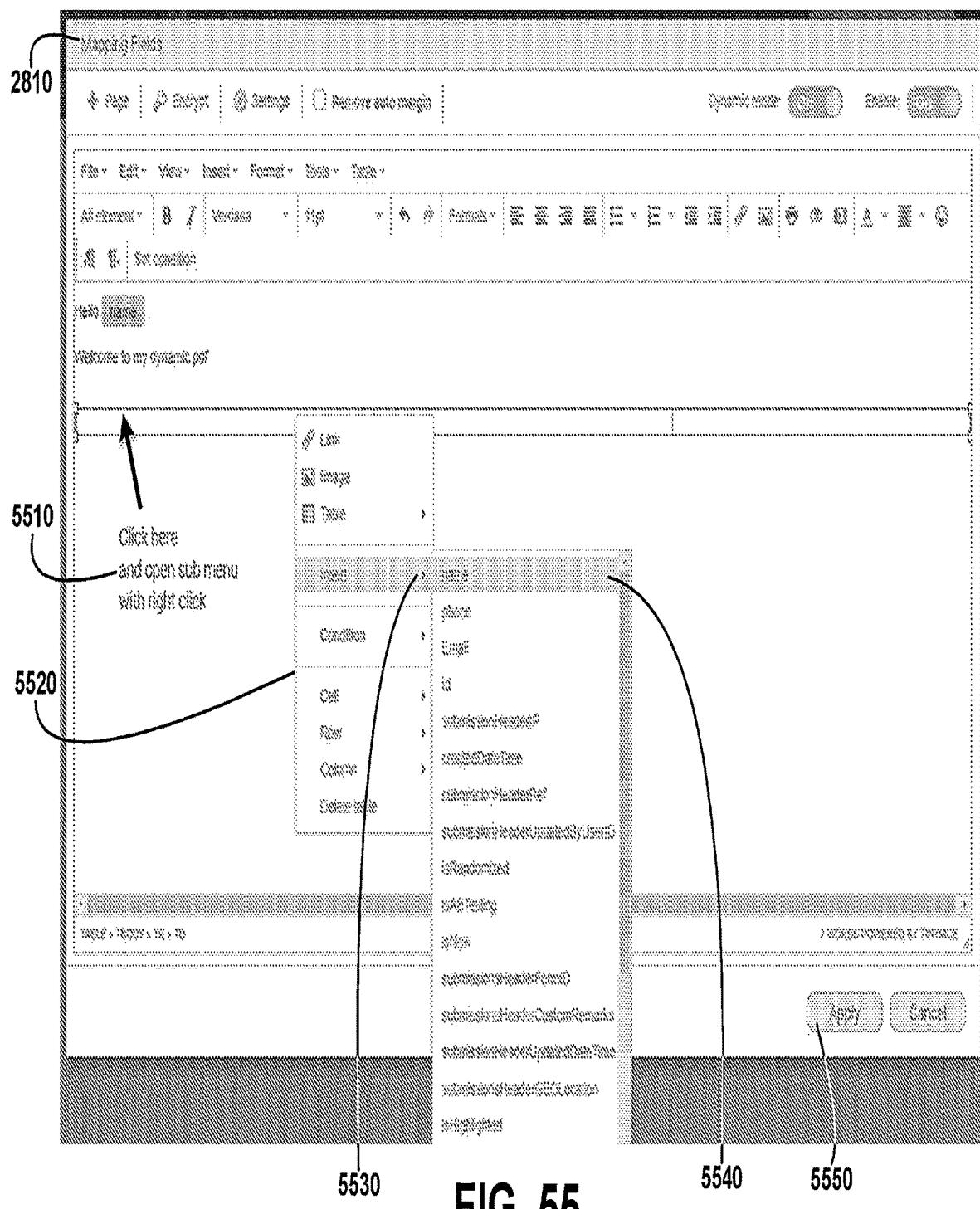
FIG. 52



**FIG. 53**



**FIG. 54**



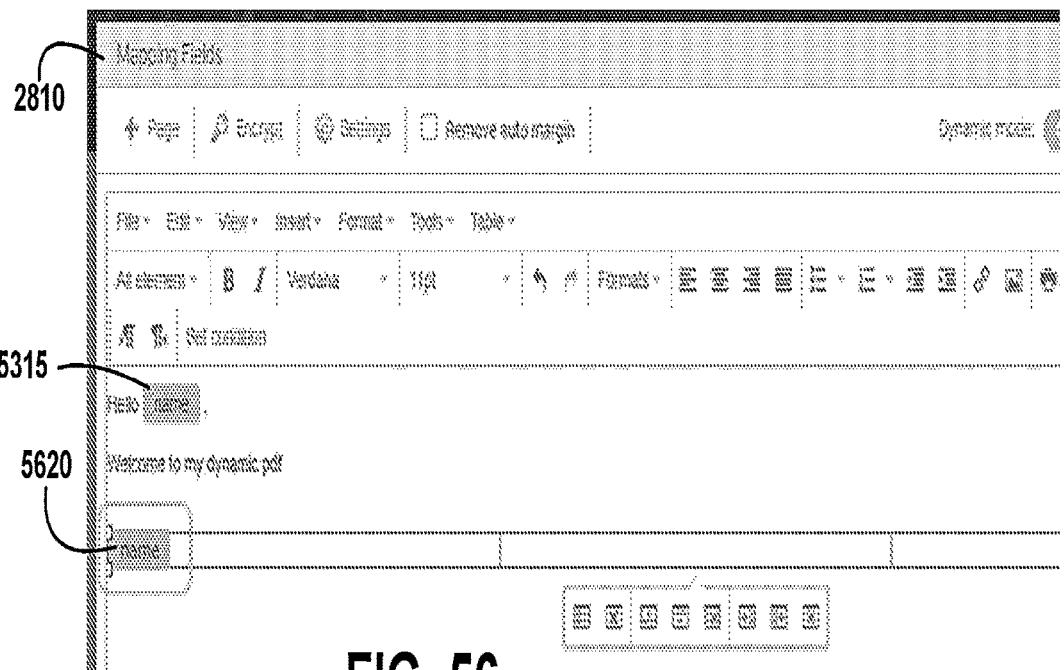


FIG. 56

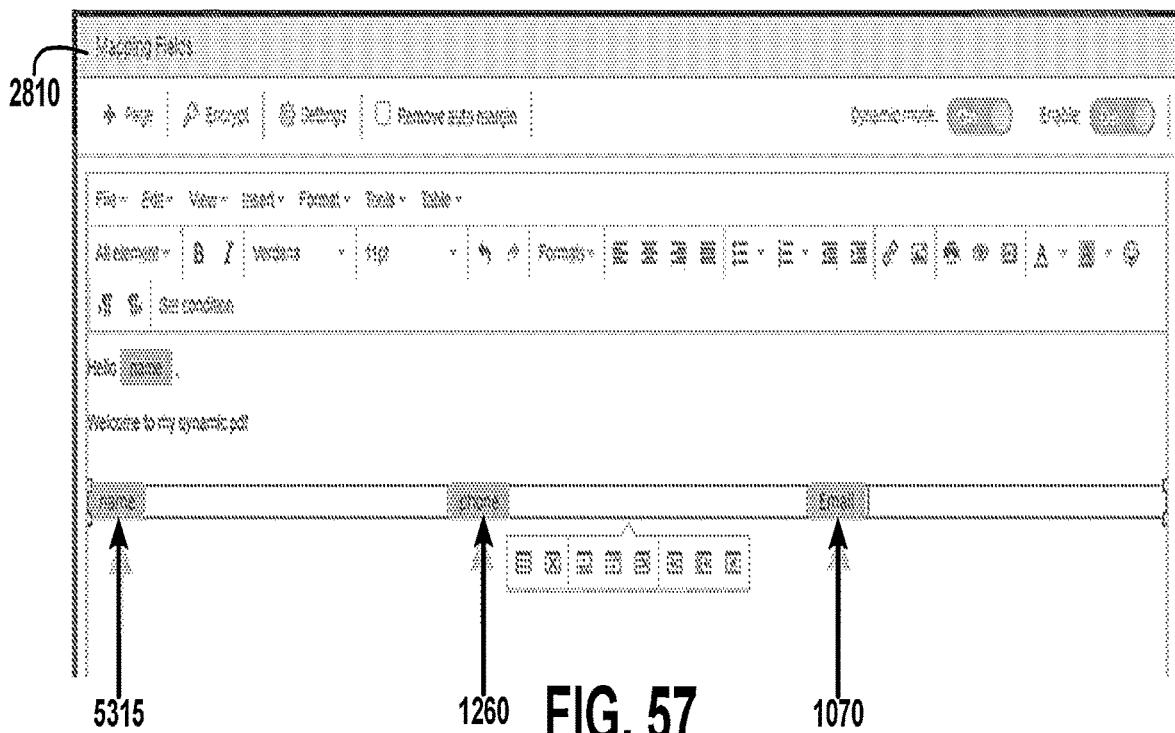


FIG. 57

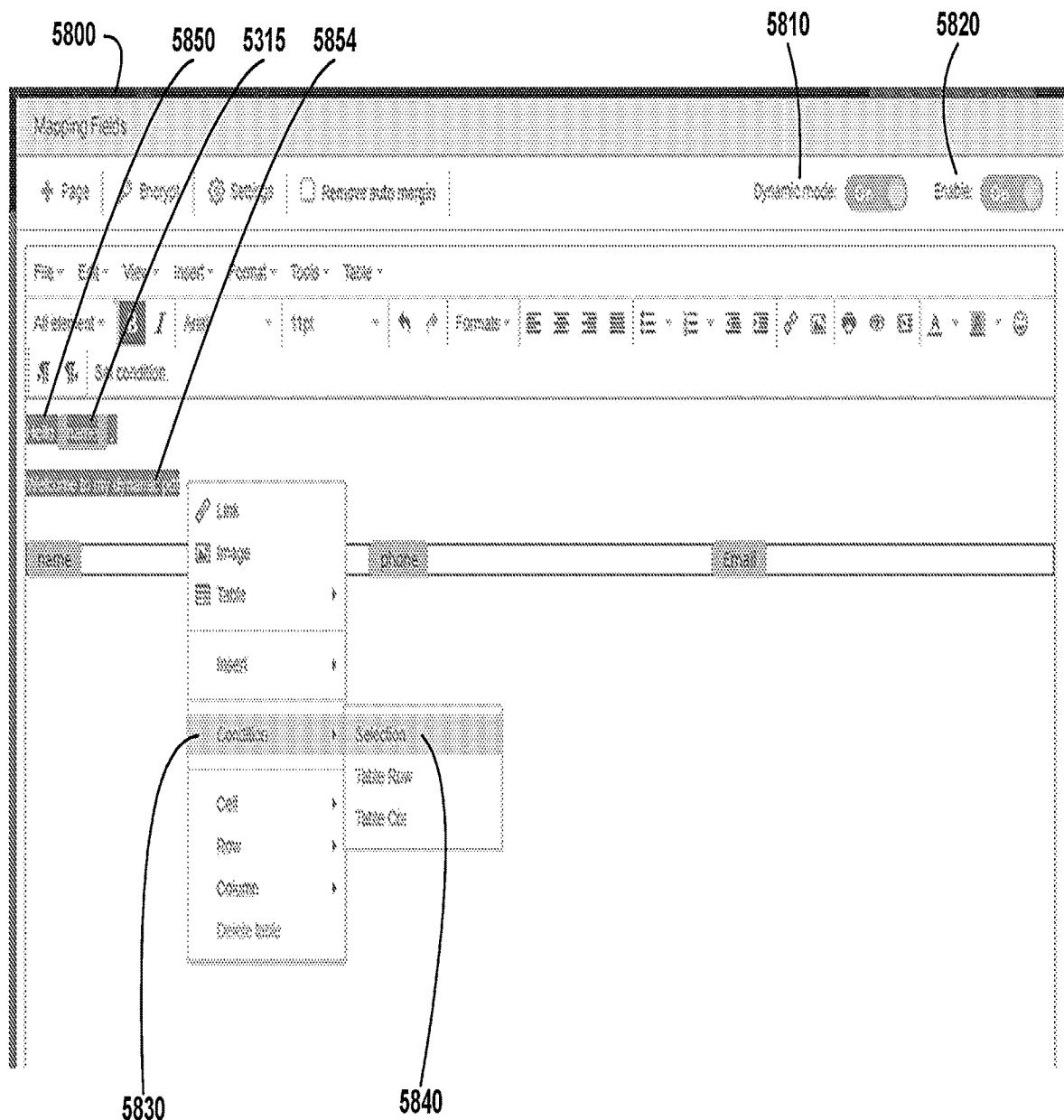


FIG. 58



FIG. 59

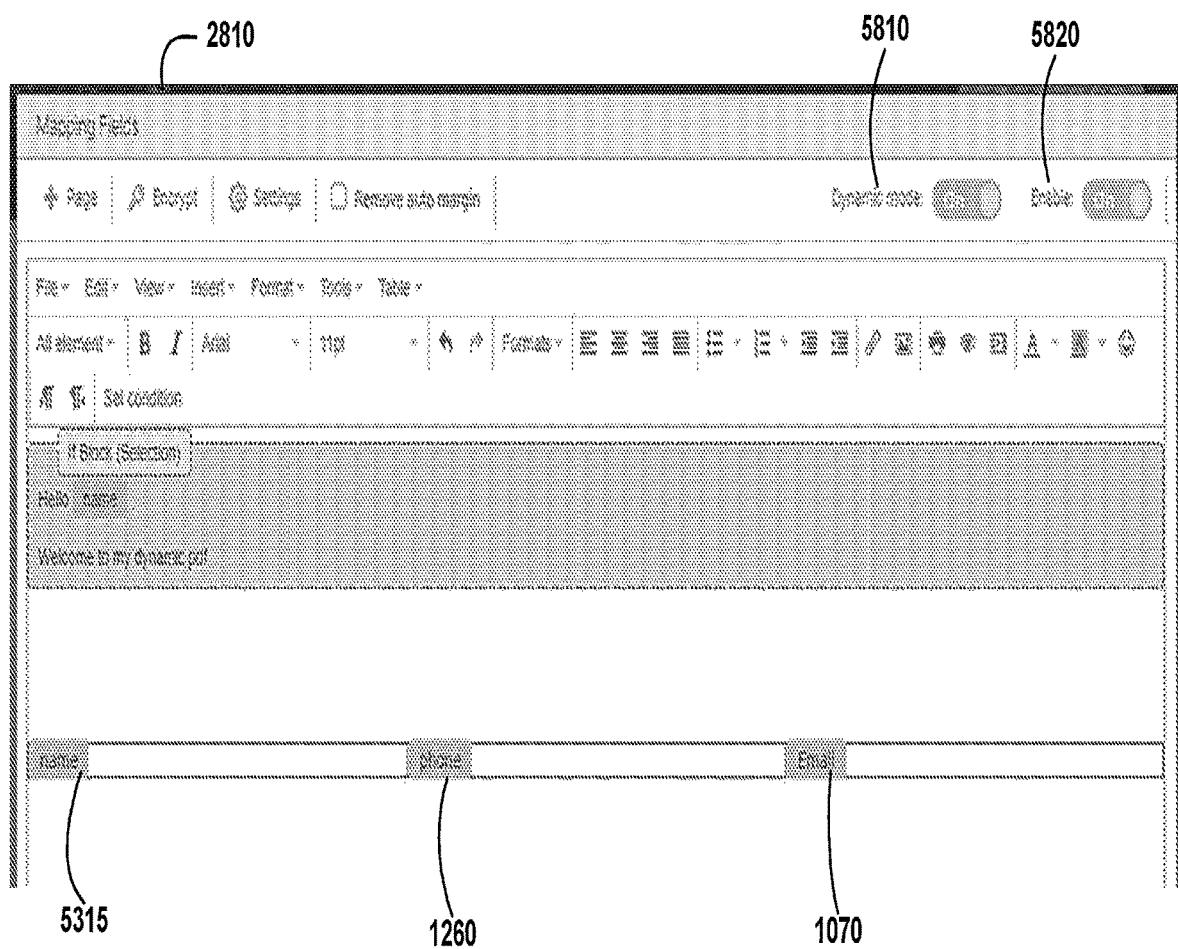
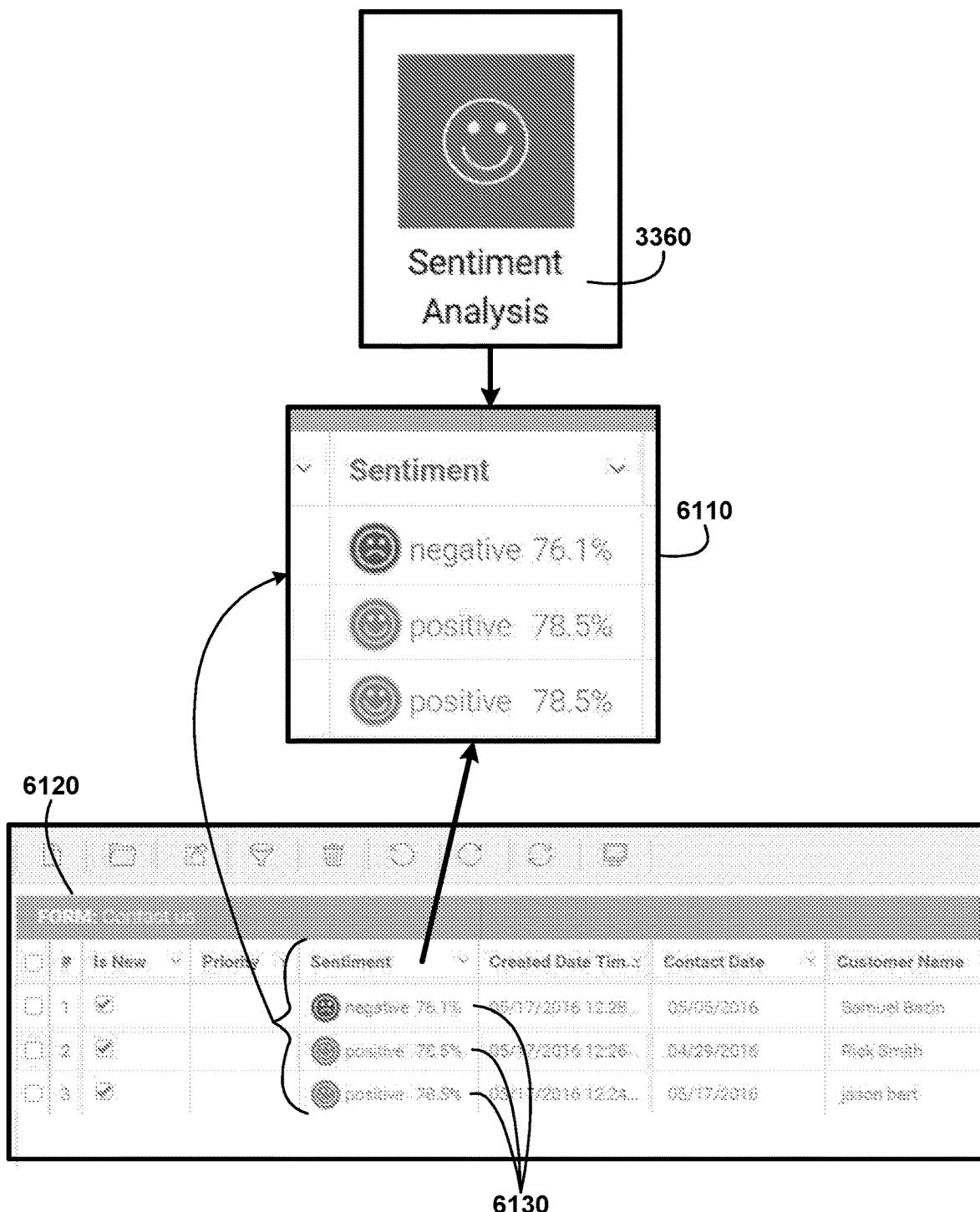
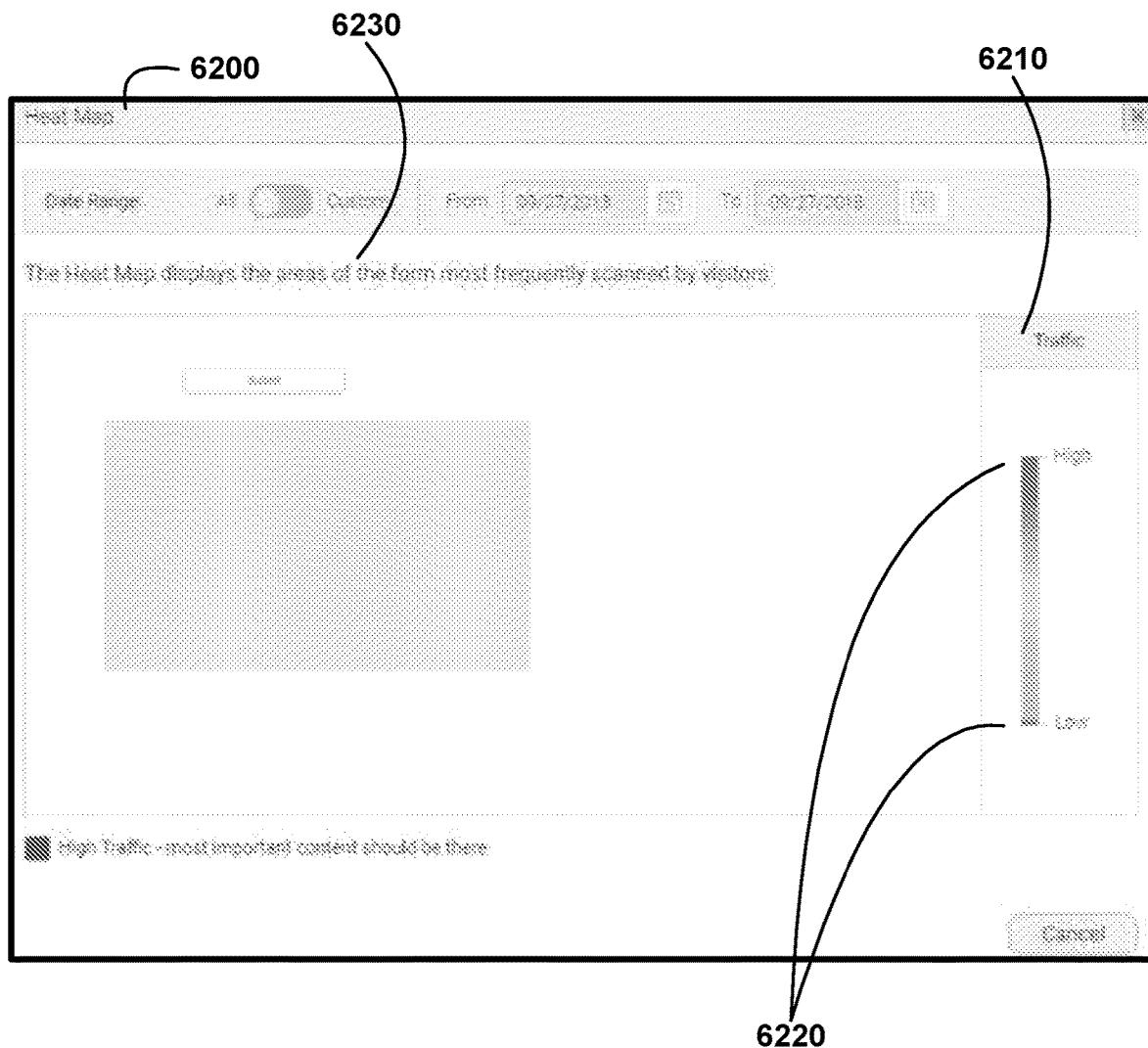


FIG. 60



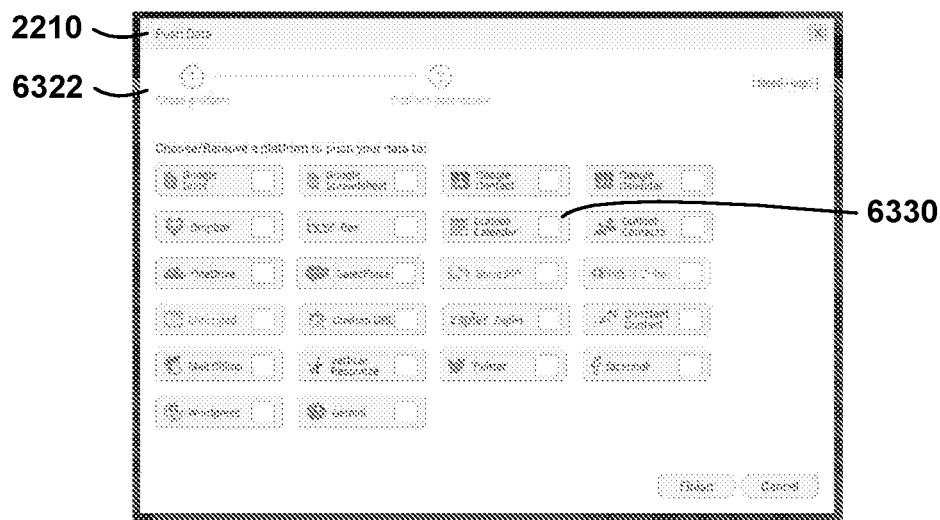
**FIG. 61**



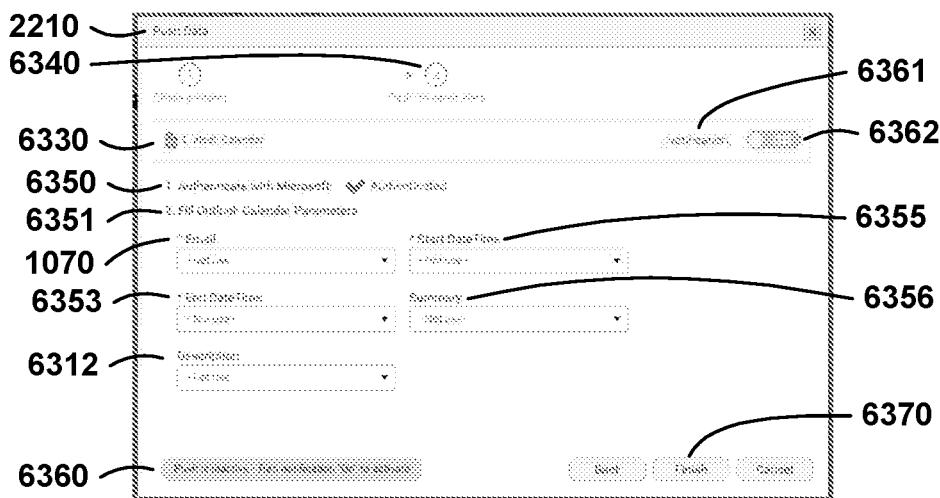
**FIG. 62**



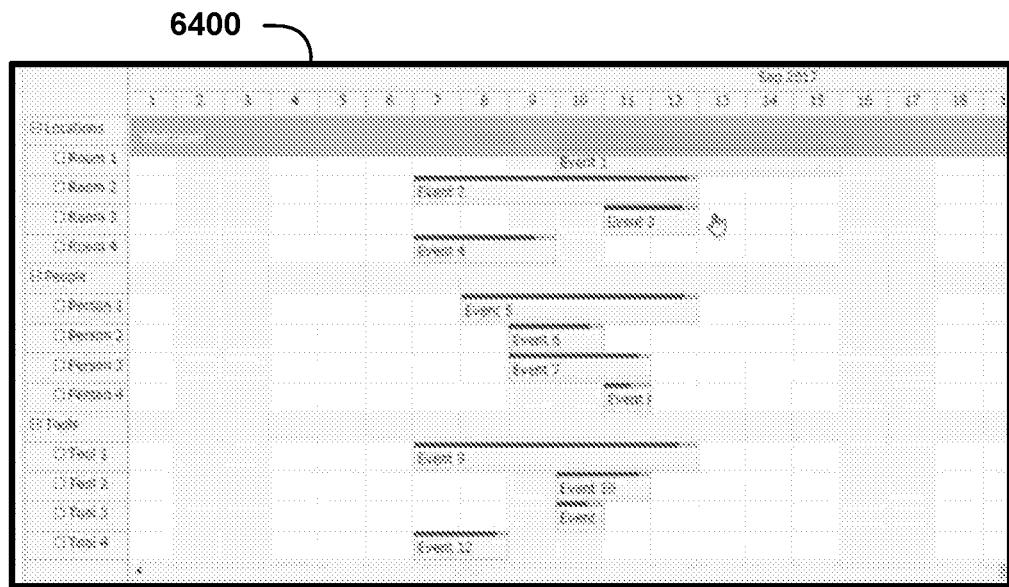
**FIG. 63A**



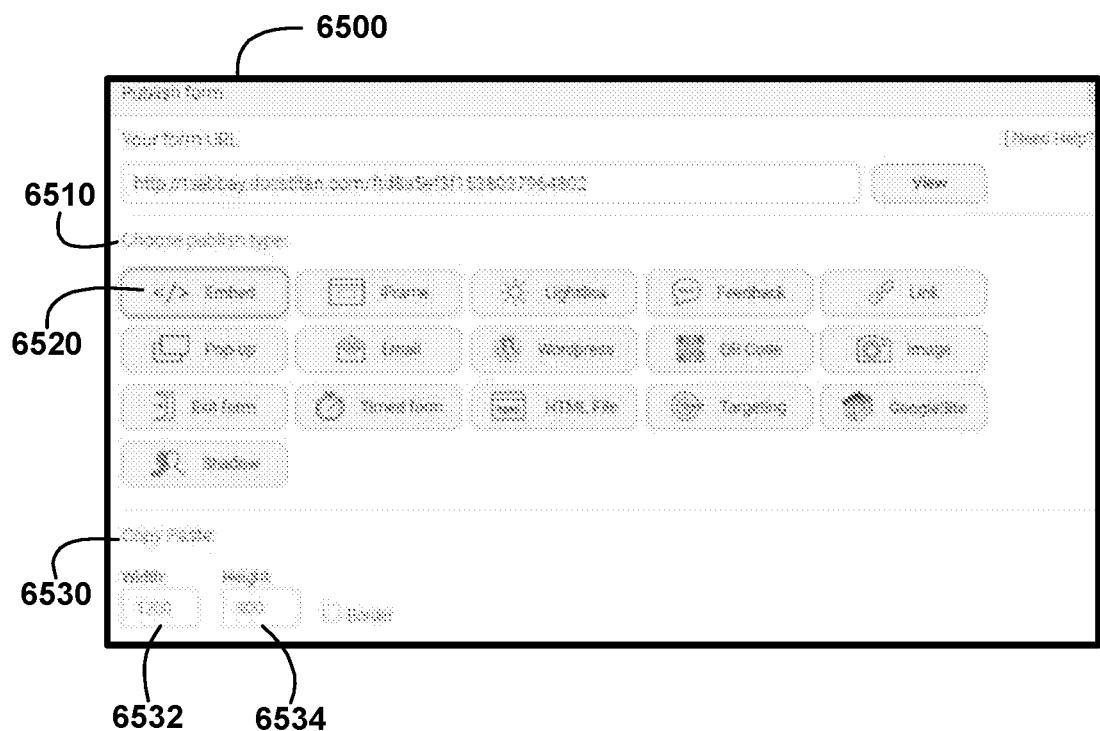
**FIG. 63B**



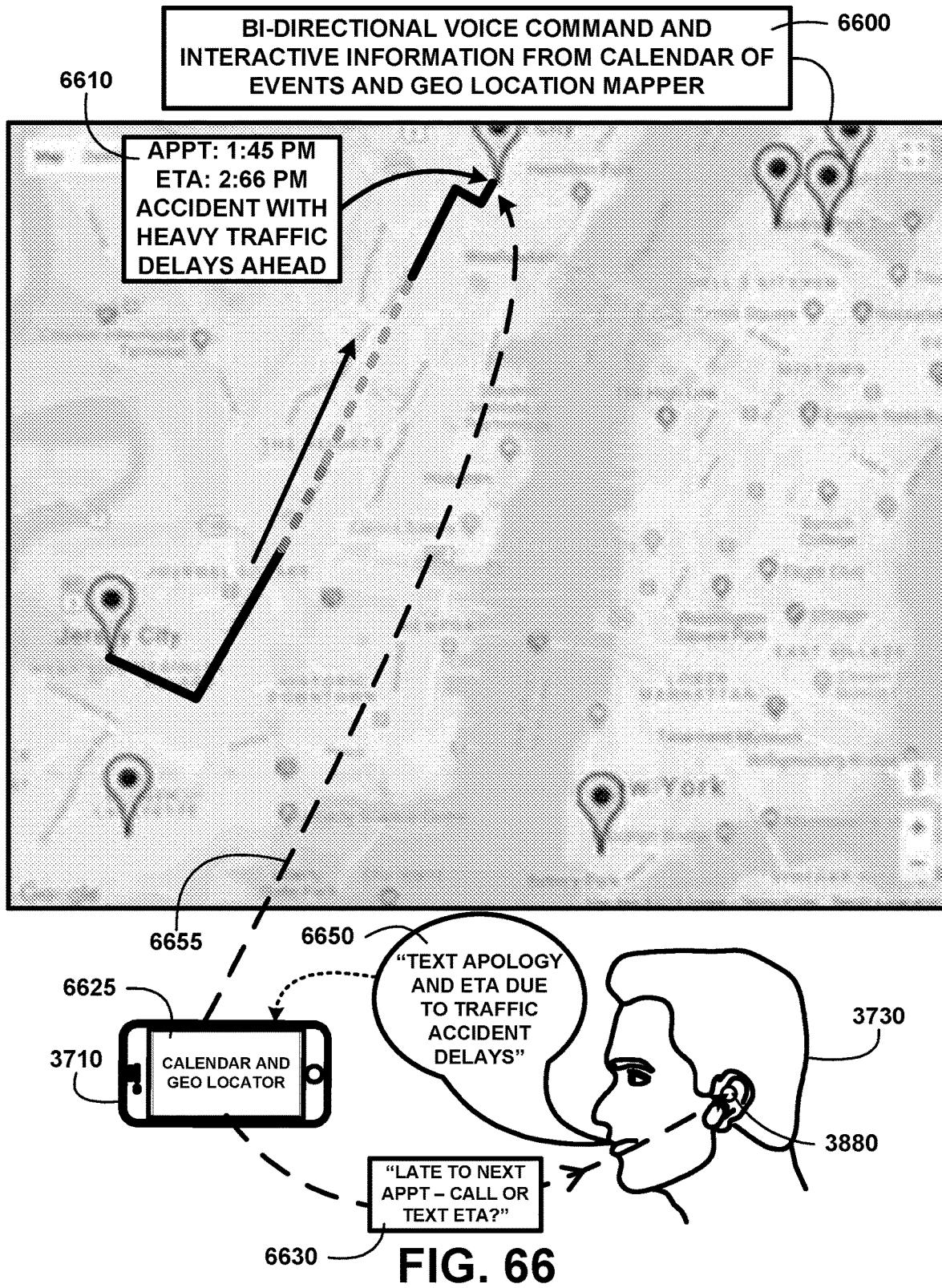
**FIG. 63C**



**FIG. 64**



**FIG. 65**



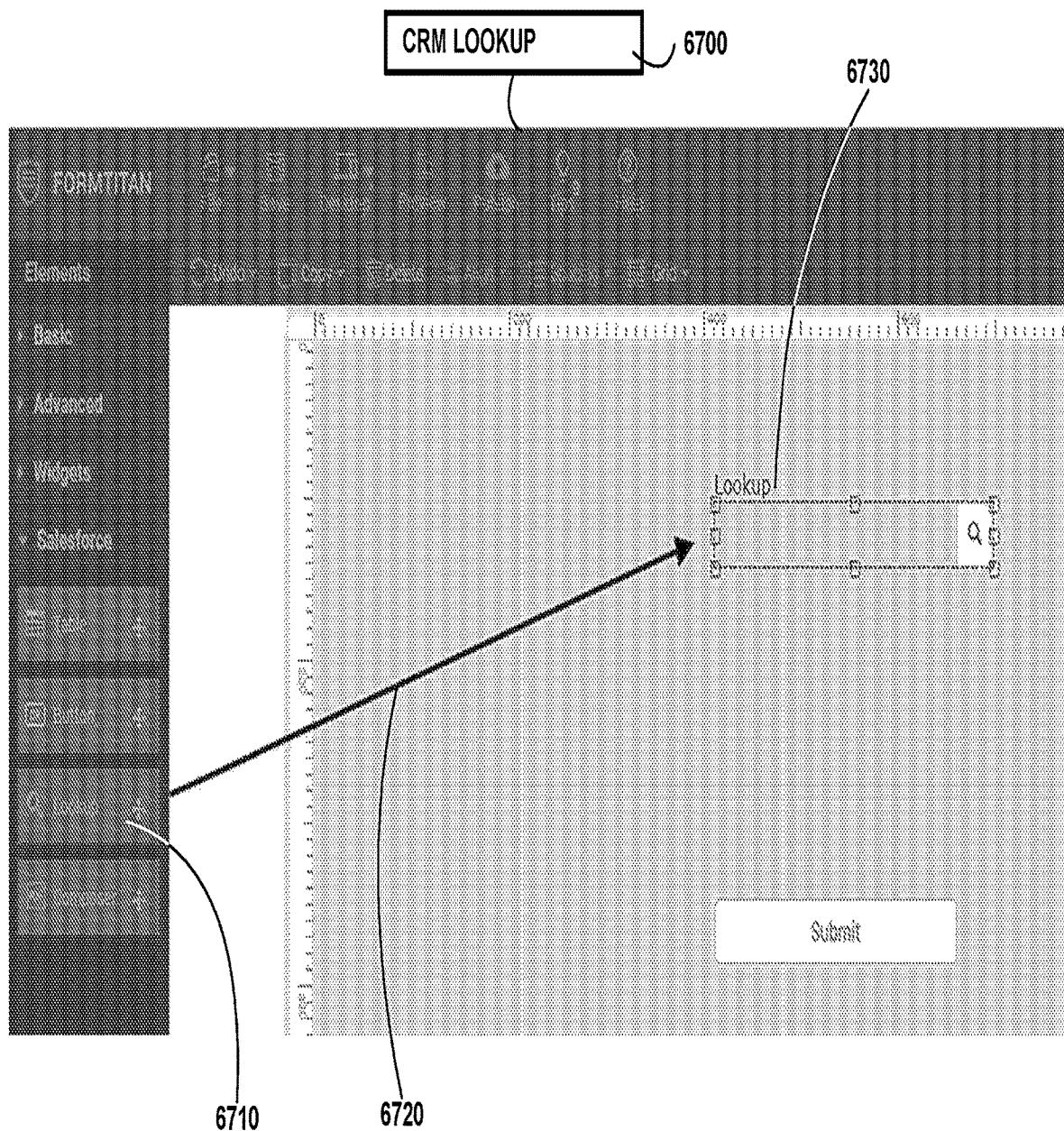


FIG. 67

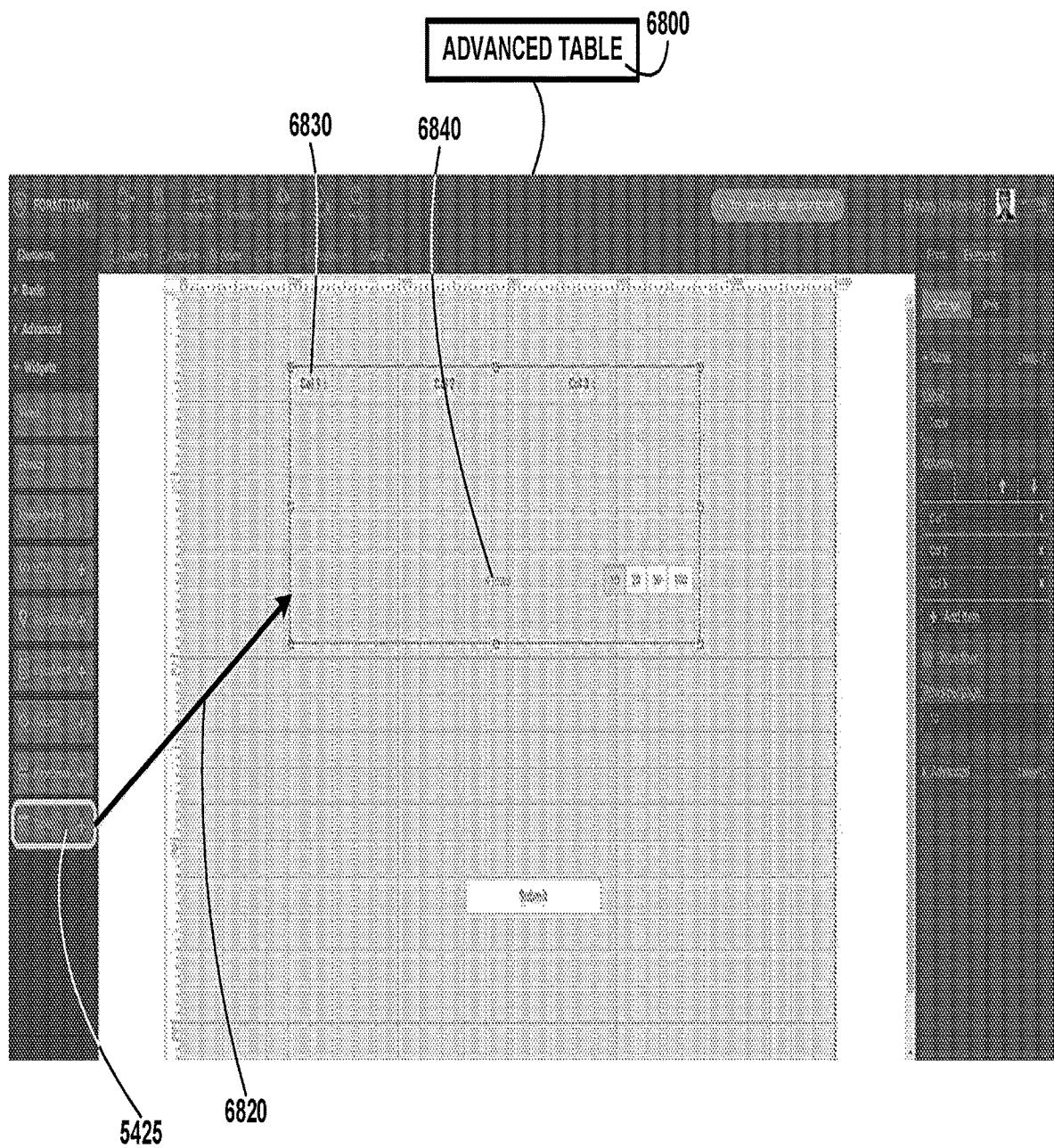


FIG. 68

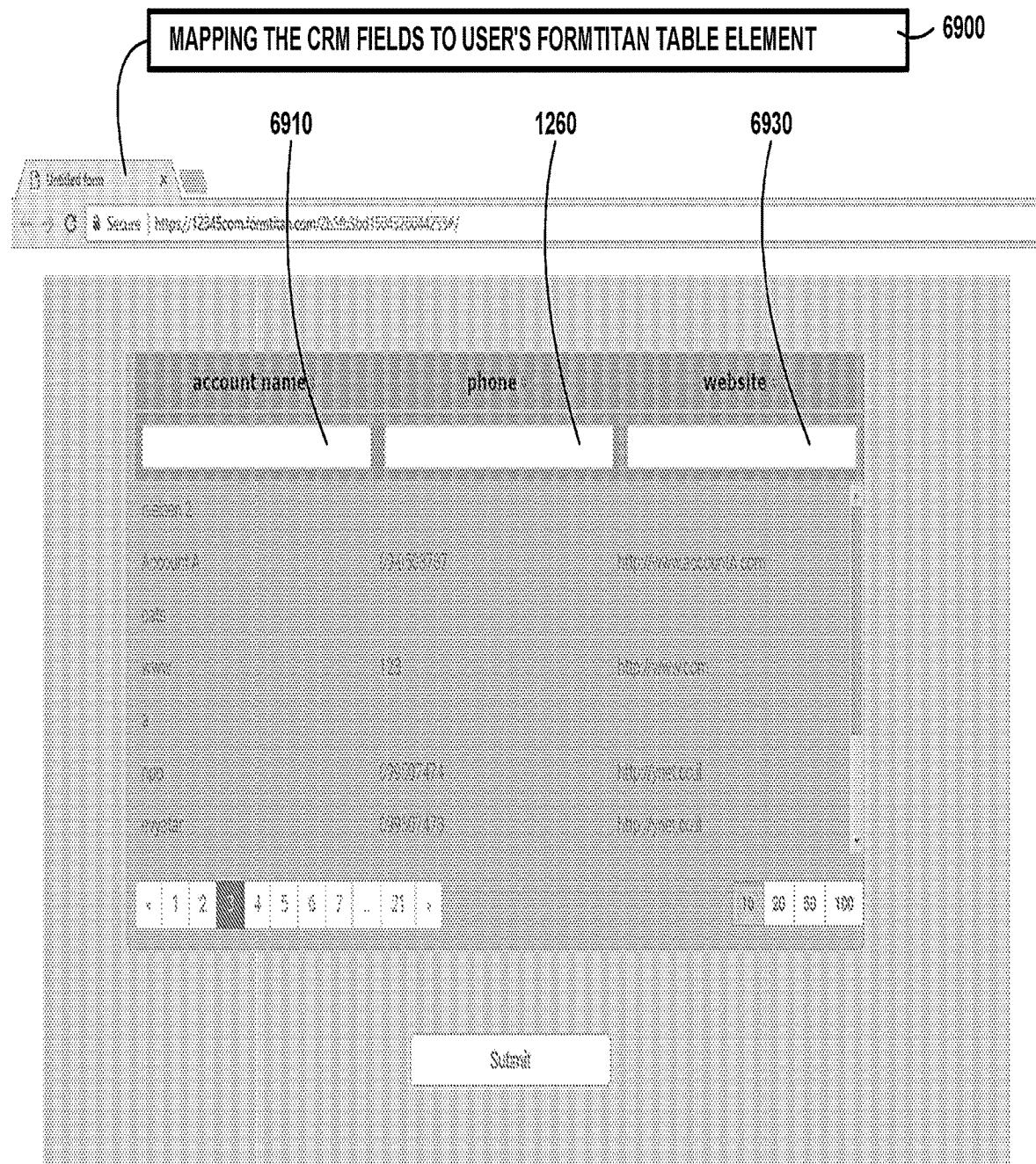
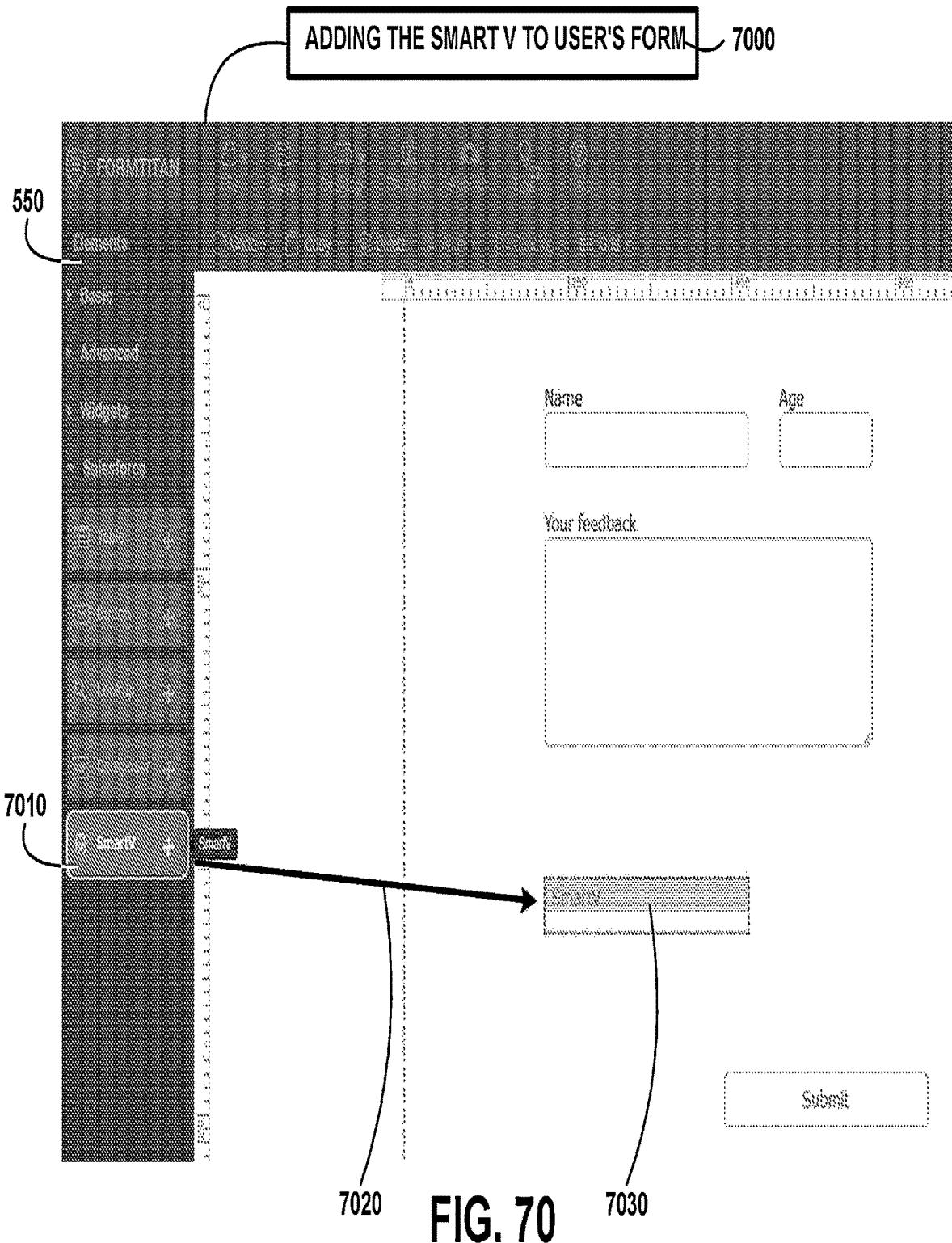


FIG. 69



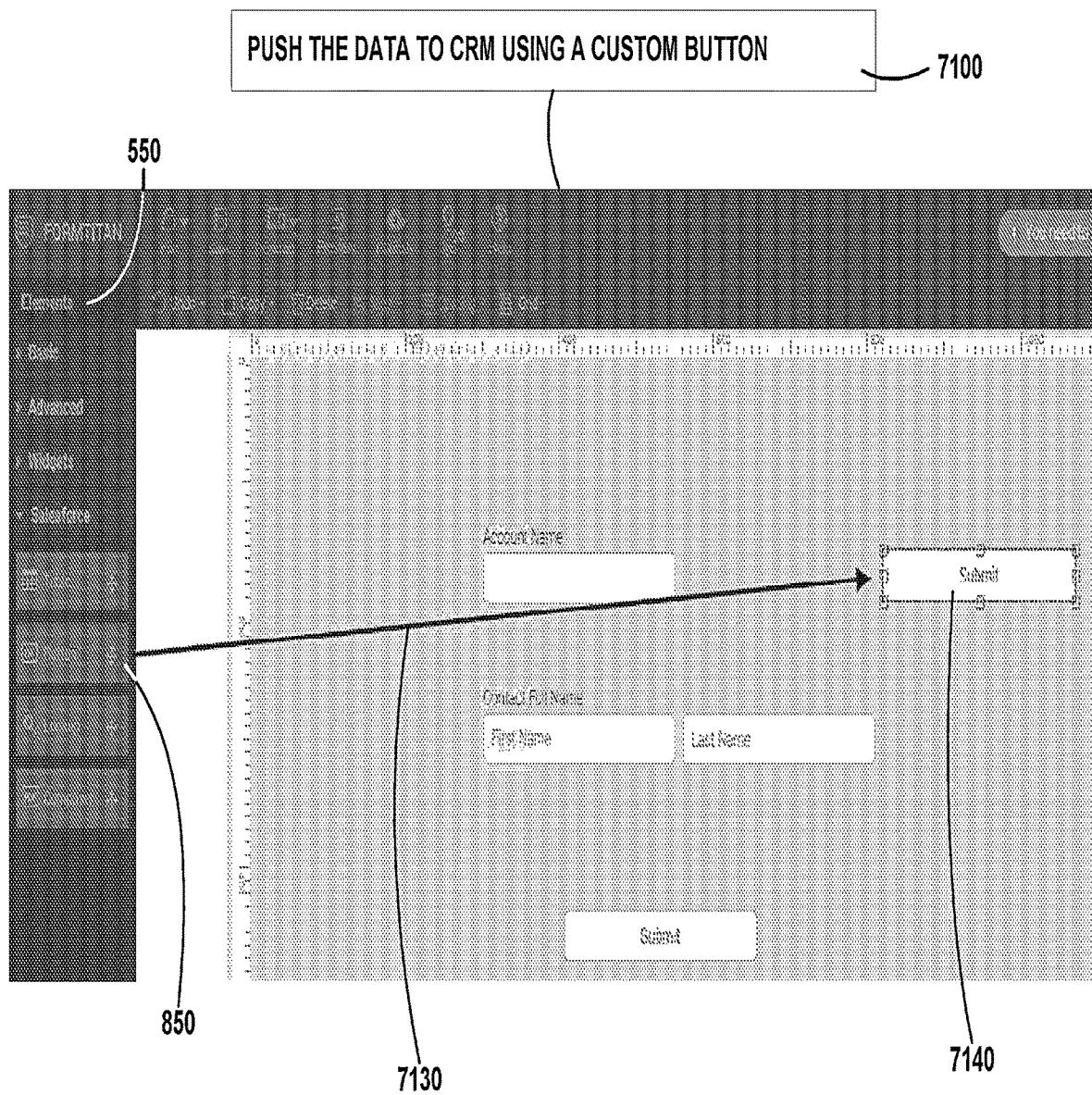


FIG. 71

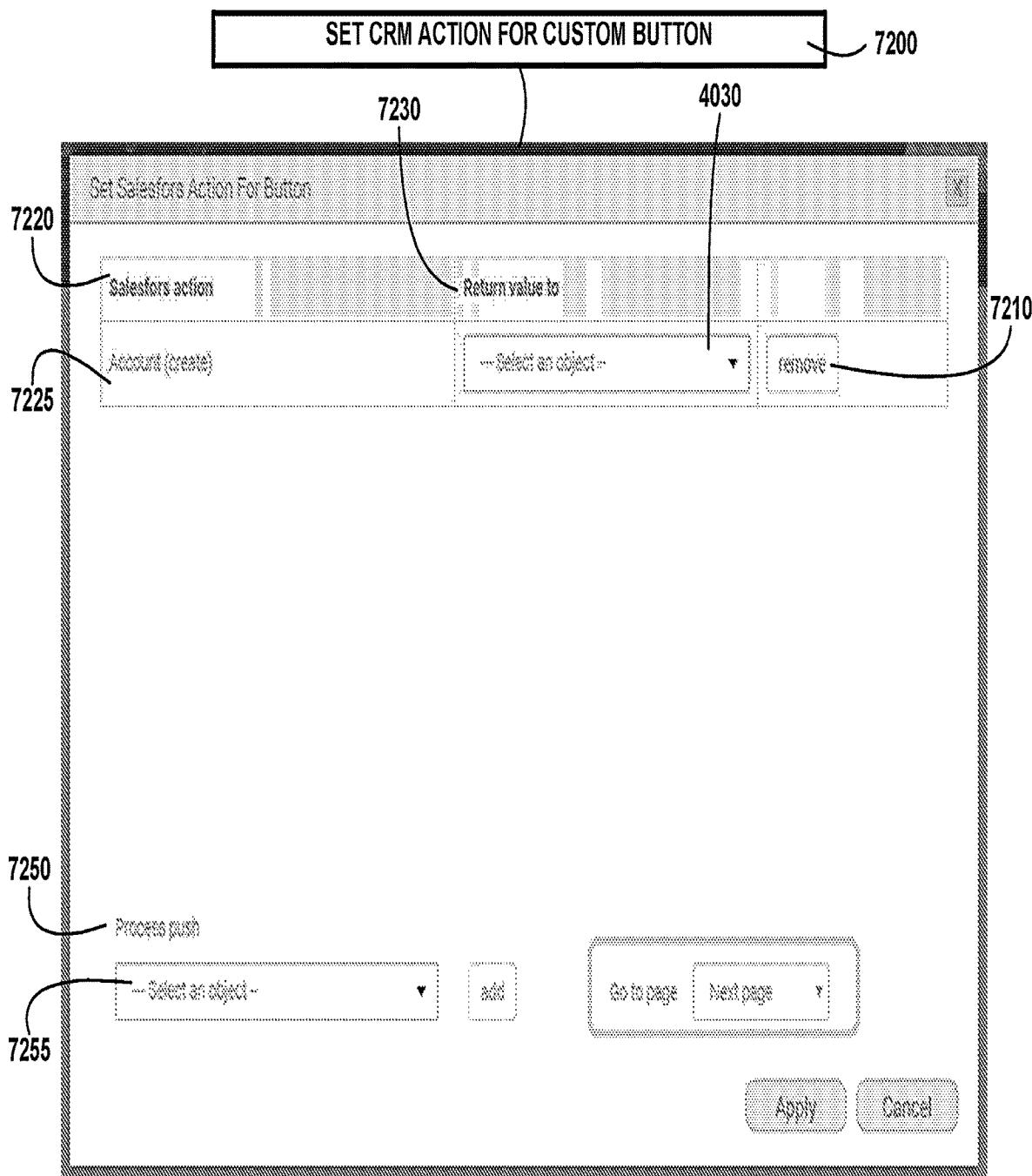


FIG. 72

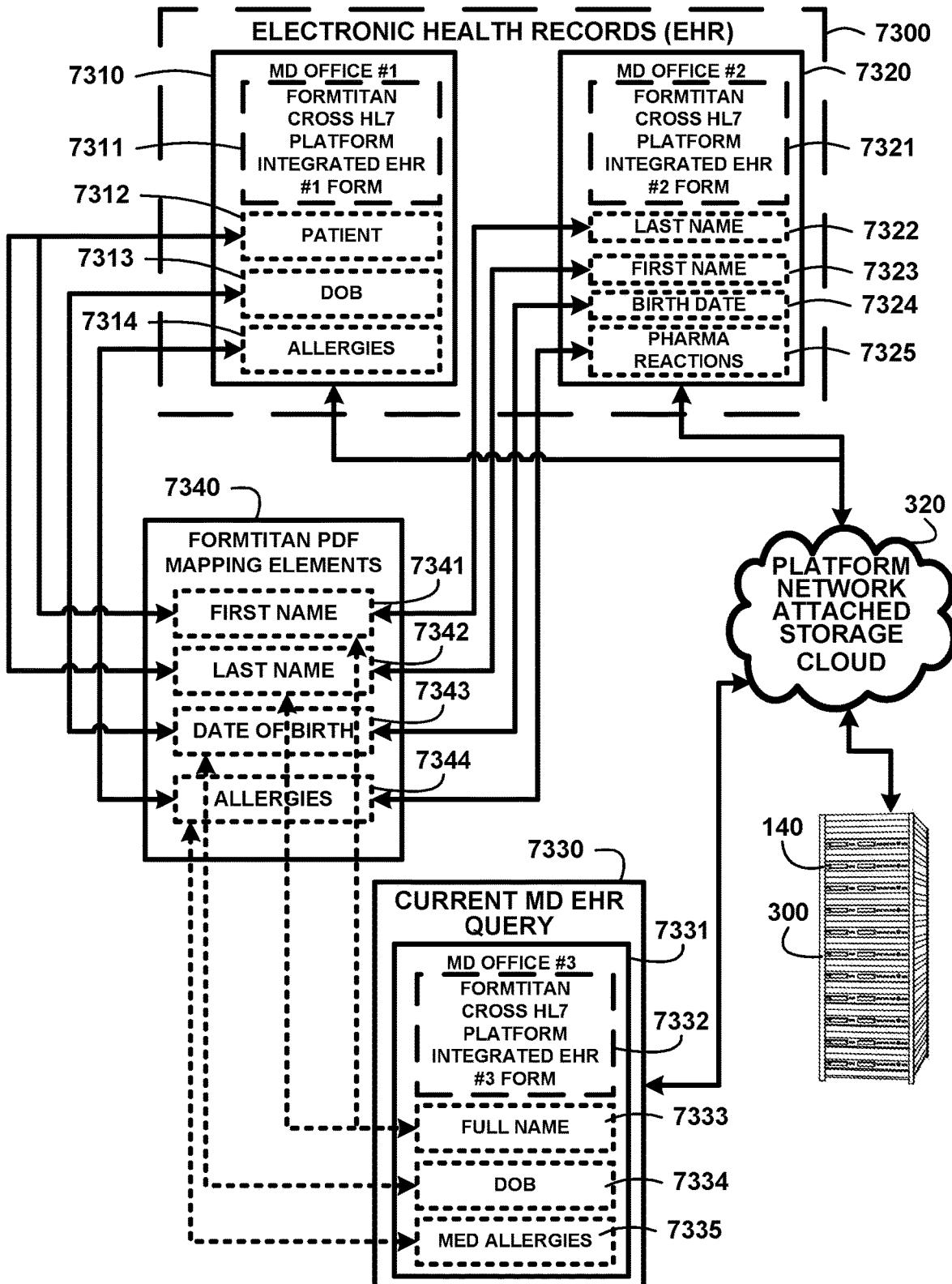
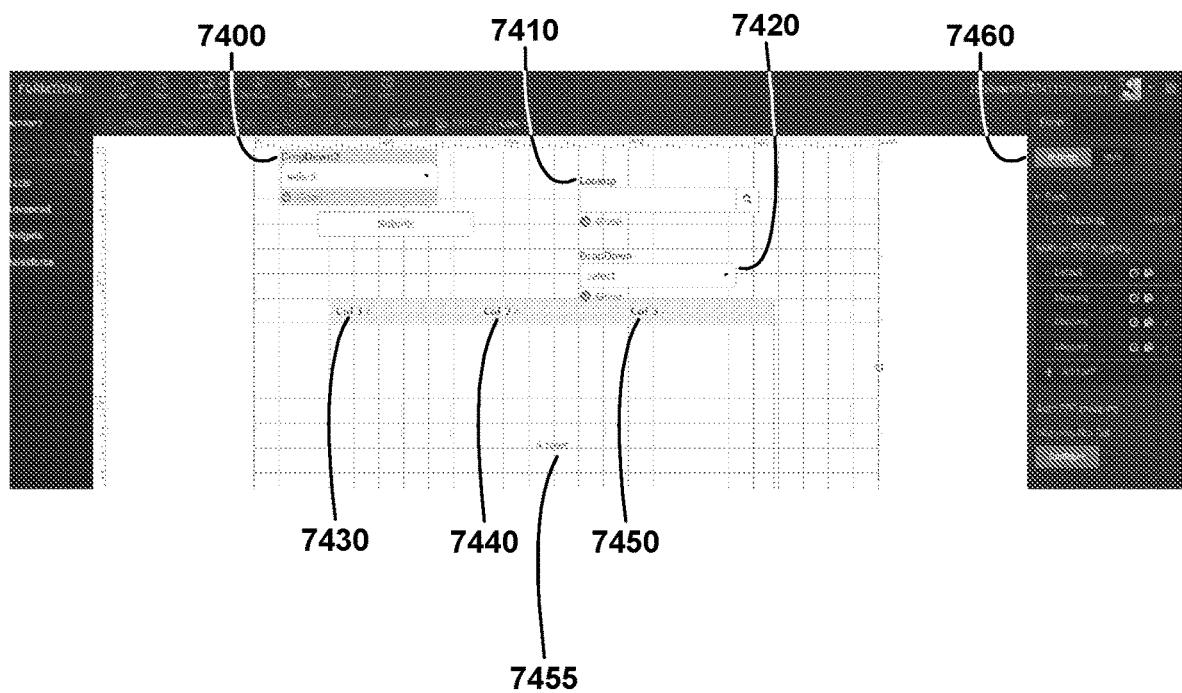


FIG. 73



**FIG. 74**

## CUSTOMIZABLE DATA MANAGEMENT FORM BUILDER METHOD AND DEVICES

### CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This patent application claims priority to United States patent application entitled: "CUSTOMIZED CUSTOMER RELATIONSHIP MANAGEMENT PLATFORM METHOD AND DEVICES", U.S. Ser. No. 62/741,489 filed on Oct. 4, 2018, the U.S. patent application being incorporated herein by reference.

### BACKGROUND

[0002] Methods and devices to create forms have been around a while. Paper forms only provide a rudimentary way to gather data that must be manually processed into a useful management and personnel tool. The advent of computer has elevated the manual task associated with form gathered data to one of more automated systems. But what has been lacking is truly user friendly systems that automate not just the data recovery and organization but expands the ease with which a user can interface with the devices and automated system to not only create a form but to gain more information from the form creation and responses the exists deeper in the content and context. The integration with other systems will assist a user in not having to duplicate the form creation task and broaden the market for gathering data and reducing the time and effort to analyze the data into useful tools for management and customer relations.

### BRIEF DESCRIPTION OF THE DRAWINGS

[0003] FIG. 1 shows a block diagram of an overview of a customized customer relationship management platform method and devices of one embodiment.

[0004] FIG. 2 shows a block diagram of an overview flow chart of a customized customer relationship management platform method and devices of one embodiment.

[0005] FIG. 3 shows for illustrative purposes only an example of the flow of data in a customized customer relationship management platform method and devices of one embodiment.

[0006] FIG. 4 shows for illustrative purposes only an example of a form builder feature of one embodiment.

[0007] FIG. 5 shows for illustrative purposes only an example of form builder uploading a user's existing pdf form of one embodiment.

[0008] FIG. 6 shows for illustrative purposes only an example of drag and drop the form builder elements over info locations on an existing user scanned paper form of one embodiment.

[0009] FIG. 7 shows for illustrative purposes only an example of a form builder pdf mapping menu of one embodiment.

[0010] FIG. 8 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode basic elements of one embodiment.

[0011] FIG. 9 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode basic text element of one embodiment.

[0012] FIG. 10 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode basic input element of one embodiment.

[0013] FIG. 11 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode basic multiple input of one embodiment.

[0014] FIG. 12 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode contact of one embodiment.

[0015] FIG. 13 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode date and time of one embodiment.

[0016] FIG. 14 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode range of one embodiment.

[0017] FIG. 15 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode video of one embodiment.

[0018] FIG. 16 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode privacy of one embodiment.

[0019] FIG. 17 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode image group of one embodiment.

[0020] FIG. 18 shows a block diagram of an overview of form builder pdf mapping dynamic mode settings of one embodiment.

[0021] FIG. 19 shows a block diagram of an overview of form builder pdf mapping dynamic mode of one embodiment.

[0022] FIG. 20 shows a block diagram of an overview of integration form builder pdf mapping Salesforce of one embodiment.

[0023] FIG. 21 shows a block diagram of an overview of integration form builder file upload—allow accept of one embodiment.

[0024] FIG. 22 shows for illustrative purposes only an example of Salesforce push integration form builder pdf mapping of one embodiment.

[0025] FIG. 23 shows for illustrative purposes only an example of push settings tab of one embodiment.

[0026] FIG. 24 shows for illustrative purposes only an example of push authentication list and status of one embodiment.

[0027] FIG. 25 shows for illustrative purposes only an example of push logs of one embodiment.

[0028] FIG. 26 shows for illustrative purposes only an example of personalized preventive care and wellness plan of one embodiment.

[0029] FIG. 27 shows for illustrative purposes only an example of HIPAA, GDPR and other compliances are supported of one embodiment.

[0030] FIG. 28 shows for illustrative purposes only an example of pdf mapping settings of one embodiment.

[0031] FIG. 29 shows for illustrative purposes only an example of pdf mapping settings general of one embodiment.

[0032] FIG. 30 shows for illustrative purposes only an example of pdf mapping settings buttons and labels of one embodiment.

[0033] FIG. 31 shows for illustrative purposes only an example of pdf mapping settings email of one embodiment.

[0034] FIG. 32 shows for illustrative purposes only an example of pdf mapping settings condition of one embodiment.

[0035] FIG. 33 shows for illustrative purposes only an example of form builder optimization features of one embodiment.

[0036] FIG. 34 shows for illustrative purposes only an example of form builder features characteristics of one embodiment.

[0037] FIG. 35 shows a block diagram of an overview of other user applications of one embodiment.

[0038] FIG. 36 shows a block diagram of an overview of other user applications enterprise environment of one embodiment.

[0039] FIG. 37A shows for illustrative purposes only an example of bi-directional voice command and interactive form building elements of one embodiment.

[0040] FIG. 37B shows for illustrative purposes only an example of bi-directional voice command and interactive form building data of one embodiment.

[0041] FIG. 38 shows for illustrative purposes only an example of bi-directional voice command and interactive information from forms of one embodiment.

[0042] FIG. 39 shows a block diagram of an overview of integration example with Salesforce of one embodiment.

[0043] FIG. 40 shows for illustrative purposes only an example of integration Salesforce object settings of one embodiment.

[0044] FIG. 41 shows for illustrative purposes only an example of integration setting Salesforce section of one embodiment.

[0045] FIG. 42 shows for illustrative purposes only an example of integration setting Salesforce fields of one embodiment.

[0046] FIG. 43A shows for illustrative purposes only an example of integration get from Salesforce of one embodiment.

[0047] FIG. 43B shows for illustrative purposes only an example of integration map Salesforce fields of one embodiment.

[0048] FIG. 44 shows for illustrative purposes only an example of integration map Salesforce object settings of one embodiment.

[0049] FIG. 45 shows for illustrative purposes only an example of integration Salesforce condition of one embodiment.

[0050] FIG. 46 shows for illustrative purposes only an example of integration Salesforce mapping corresponding fields of one embodiment.

[0051] FIG. 47 shows for illustrative purposes only an example of integration Salesforce mapping contact of one embodiment.

[0052] FIG. 48 shows for illustrative purposes only an example of integration Salesforce get contact of one embodiment.

[0053] FIG. 49 shows for illustrative purposes only an example of integration Salesforce email of one embodiment.

[0054] FIG. 50 shows for illustrative purposes only an example of pdf block condition elements of one embodiment.

[0055] FIG. 51 shows for illustrative purposes only an example of pdf block condition dynamic mode of one embodiment.

[0056] FIG. 52 shows for illustrative purposes only an example of pdf block condition pdf settings of one embodiment.

[0057] FIG. 53 shows for illustrative purposes only an example of pdf block condition mapping fields all elements of one embodiment.

[0058] FIG. 54 shows for illustrative purposes only an example of pdf block condition mapping fields table of one embodiment.

[0059] FIG. 55 shows for illustrative purposes only an example of pdf block condition starting mapping fields of one embodiment.

[0060] FIG. 56 shows for illustrative purposes only an example of pdf block condition mapping field name of one embodiment.

[0061] FIG. 57 shows for illustrative purposes only an example of pdf block condition mapping fields of one embodiment.

[0062] FIG. 58 shows for illustrative purposes only an example of pdf block condition mapping field selections of one embodiment.

[0063] FIG. 59 shows for illustrative purposes only an example of pdf block rule condition selection of one embodiment.

[0064] FIG. 60 shows for illustrative purposes only an example of pdf block condition mapping field dynamic mode settings of one embodiment.

[0065] FIG. 61 shows for illustrative purposes only an example of a sentiment analyze feature of one embodiment.

[0066] FIG. 62 shows for illustrative purposes only an example of heat map integration of one embodiment.

[0067] FIG. 63A shows for illustrative purposes only an example of integrating a form platform into form builder of one embodiment.

[0068] FIG. 63B shows for illustrative purposes only an example of authenticate a form platform of one embodiment.

[0069] FIG. 63C shows for illustrative purposes only an example of choosing form fields data to push to form platform of one embodiment.

[0070] FIG. 64 shows for illustrative purposes only an example of an event calendar feature of one embodiment.

[0071] FIG. 65 shows for illustrative purposes only an example of a publish form feature of one embodiment.

[0072] FIG. 66 shows for illustrative purposes only an example of a voice interactive calendar and geo locator of one embodiment.

[0073] FIG. 67 shows for illustrative purposes only an example of Salesforce lookup of one embodiment.

[0074] FIG. 68 shows for illustrative purposes only an example of advanced table of one embodiment.

[0075] FIG. 69 shows for illustrative purposes only an example of mapping the Salesforce fields to a FormTitan table element of one embodiment.

[0076] FIG. 70 shows for illustrative purposes only an example of adding the Smart V to a form of one embodiment.

[0077] FIG. 71 shows for illustrative purposes only an example of push the data to Salesforce using a custom button of one embodiment.

[0078] FIG. 72 shows for illustrative purposes only an example of set Salesforce action for custom button of one embodiment.

[0079] FIG. 73 shows for illustrative purposes only an example of an HL7 EHR cross platform application of one embodiment.

[0080] FIG. 74 shows for illustrative purposes only an example of a dropdown2 form settings element of one embodiment.

#### DETAILED DESCRIPTION OF THE INVENTION

[0081] In a following description, reference is made to the accompanying drawings, which form a part hereof, and in which is shown by way of illustration a specific example in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

##### General Overview:

[0082] It should be noted that the descriptions that follow, for example, in terms of customized customer relationship management platform method and devices is described for illustrative purposes and the underlying system can apply to any number and multiple types of application uses. In one embodiment of the present invention, the customized customer relationship management platform method and devices can be configured using a user existing pdf form to overlay interactive pdf mapping elements to create an html interactive pdf form. The customized customer relationship management platform method and devices can be configured to include creating a new html interactive pdf form using a dynamic pdf mapping device and can be configured to include creating an html form using an auto pdf mapping device. The customized customer relationship management platform method and devices can be configured to include integration of the html interactive pdf form created with third party customer relationship management and other types of platforms using the present invention.

[0083] “Customized customer relationship management platform method and devices”, “Customized customer relationship management platform”, “Customized customer relationship management” are terms used herein and can additionally be expressed without any change in meaning as “FormTitan” in any case lower, upper or mixed.

[0084] The phrase “form filler” herein means a user filling out entries in a customized customer relationship management form builder created form.

[0085] Customized customer relationship management platform method and devices create a powerful cloud based platform for creating online forms, applications, landing pages and portals. Designed for any type of business ranging from small nonprofits to large enterprises. Customized customer relationship management platform method and devices offers Smart integrations, conditional logic, document generation, payments, Lead generation, surveys as well as Innovative bidirectional Salesforce and dynamic CRM integrations. Smart integrations, bidirectional Salesforce and dynamic CRM integrations also include settings including auto save/push, auto pull, auto translate, custom translate, and phone survey.

[0086] Auto save/push and auto pull selections for example when selected in the settings will automatically save the data entered in the FormTitan customized form but also simultaneously in the CRM integrated system for example Salesforce without the user having to enter any additional commands. The auto function feature of push and pull will enter, delete and edit any added or edited data

automatically without the user having to enter any additional commands on both forms in both FormTitan and the CRM integrated system including Salesforce using the bidirectional feature.

[0087] Auto translate and custom translate settings will allow the user to build their form in a selected language and an online form that needs to be displayed in several languages, can use our Auto Translate optimization to easily have it presented to foreign users in their native language. In some cases a user may find the automatic translation insufficient or not precise enough for technical terms, or even simple words with multiple meanings and need to translate things manually. When using custom translate settings a user can control all the labels, captions, user tips, button text in a form, and can create different translations to as many languages as desired. Other features used in translation include Custom Translation allows Paragraph manual translation, and Translation for Table mandatory message—added to captions. HIPAA, GDPR and other compliances are supported as well of one embodiment.

[0088] The phone survey feature allows a user to use account phone numbers from a simple table they have created that includes account phone numbers for example to map for example Salesforce country and State picklists of the account phone numbers. The mapped account phone numbers can then be used to conduct a phone survey on a user selected service or product to determine the picklist accounts sentiments and opinions of the selected service or product of one embodiment.

[0089] FIG. 1 shows a block diagram of an overview of a customized customer relationship management platform method and devices of one embodiment. FIG. 1 shows a block diagram of an overview of a customized customer relationship management platform method and devices of one embodiment. FIG. 1 shows a customized customer relationship management platform 100. The customized customer relationship management platform method and devices is a user interface to create interactive pdf forms for use with user applications 170. The interactive pdf forms are created by a user without the use of coding using the features and elements of the customized customer relationship management platform method and devices to build forms that user clients and other can readily use to enter data that is accessible to the user.

[0090] Customized customer relationship management platform method and devices can integrate other platforms including customer relationship management (CRM) and other types of platforms to populate the other platforms with the customized customer relationship management platform method and devices created pdf forms quickly and productively for their applications. The customized customer relationship management platform 100 includes features including a form builder 110 with pdf mapping 130 and cross platform integration 140. The customized customer relationship management platform 100 includes a digital process to convert to minimode 120 to display on digital devices 121. The customized customer relationship management platform 100 provides processes for a user to create interactive pdf forms from user existing pdf forms 150, create interactive user custom pdf forms 155, and integrate user CRM platforms and other user platforms 160 into the customized customer relationship management platform 100 to facilitate the use for user applications 170 of one embodiment.

[0091] FIG. 1 shows a customized customer relationship management platform method and devices customized customer relationship management platform **100**. The customized customer relationship management platform **100** includes a form builder **110** feature. The form builder feature includes pdf mapping **130** for converting user existing pdf forms into responsive PDF forms and for creating new responsive user custom pdf forms **155**. The form builder **110** feature includes cross platform integration **140** to allow a user to use the form builder created forms on 3<sup>rd</sup> party user CRM platforms and other user platforms.

[0092] Access to the customized customer relationship management platform **100** can be customized to display without losing any functionality to a display on digital devices a user chooses with a feature to convert to minimode **120**. The mini mode feature is used to convert the function access to a size that fits the screen size of the user digital device. The customized customer relationship management platform **100** includes a number of different types of forms or user's activities that can be configured into a form format can be created for use on a wide range of user applications **170**.

[0093] The customized customer relationship management platform **100** includes a number of features including the Form Builder **100**, Account features, Form optimization, Push Integrations, Form design, Data Collection, Data analysis and others. The customized customer relationship management platform **100** includes function products for example for Business including, PDF Mapping **130**, Targeting and Workflow, functions for Optimization including CRO, A/B Testing, Auto Translate for multi-lingual conversions, and Online Chat and functions for customer relationship management (CRM) including for example Salesforce, Dynamic 365 and others. A user can use the Online Chat to get assistance and other help functions including Q&A for asking us questions is the fastest and most effective way to get assistance, FAQ for answers to commonly asked questions, TOPICS including a complete index of the FormTitan topics organized by categories, FEATURES for greater detail in the topics directory to find the feature being sought for and simply send in an inquiry of one embodiment.

[0094] FIG. 1 shows a customized customer relationship management platform **100**. The customized customer relationship management platform **100** provides a website interface within the customizable application builder with no code visualization tools for an application creation user. The customized customer relationship management platform **100** website interface includes a form builder **110** feature for an application creation user to use for creating at least one form. The form builder **110** feature is one component of the no code visualization tools forming the customized customer relationship management platform **100**. The form builder **110** feature component with at least one form element for receiving form input from an end user, each associated with a conditional rule builder, an auto pdf mapping tool, a manual pdf mapping tool and at least two dynamic pdf mapping tools. Upon submission of the form by the end user the auto pdf mapping tool automatically generates a pdf of the entire form, including the form element and the form input from the end user of one embodiment.

[0095] The manual pdf mapping tool includes no code visualization tools that allows the application creation user to drag and drop form elements onto a predetermined pdf template file, wherein upon submission of the form, the

manual pdf mapping tool automatically generates a customizable pdf of the form that includes the customizable form elements with the form input from the end user. The dynamic pdf mapping tool includes no code visualization tools that allows the application creation user to drag and drop form elements onto a blank html template or edit the html of the blank html template. Upon submission of the form, the dynamic pdf mapping tool dynamically generates a customizable pdf of the form that includes the customizable form elements with the form input from the end user of one embodiment.

#### Overview Flow Chart:

[0096] FIG. 2 shows a block diagram of an overview flow chart of a customized customer relationship management platform method and devices of one embodiment. FIG. 2 shows creating a customer relationship management platform network **210** and creating a customer relationship management platform network attached storage cloud device **200**. Operating a customized customer relationship management platform **220** includes using a customized form builder **230** with features for using pdf mapping **231**, using a dynamic mode **232** and using an auto mode **233**. Using a customized form builder **230** includes providing cross platform integration **234** employing a get feature **235** and employing a push/pull feature **236**. Converting user interactivity to a minimode **240** for displaying on digital devices **241** includes a computer **242**, a tablet **243**, a laptop computer **244** and a smart phone **245**. Using a customized form builder **230** includes creating interactive pdf forms from user existing pdf forms **150** and user custom pdf forms **155**. Using a customized form builder **230** includes integrating user CRM platforms and other user platforms **160** for performing customized user applications **260** of one embodiment.

[0097] FIG. 2 shows creating a customized customer relationship management platform network and creating a customized customer relationship management platform network attached storage cloud device. Operating a customized customer relationship management platform providing an interactive interface with the customized customer relationship management platform network and customized customer relationship management platform network attached storage cloud device.

[0098] Operating a customized customer relationship management platform includes using a customized form builder for using pdf mapping to create custom responsive PDF forms using a dynamic mode and using an auto mode and converting a user's scanned existing paper form into a responsive PDF form. The customized customer relationship management platform is providing cross platform integration employing a get feature and employing a push/pull feature to apply the form builder created forms on a user CRM platforms and other user 3<sup>rd</sup> party platforms.

[0099] Operating a customized customer relationship management platform a user may use many types of digital devices. The display screen for a computer is much larger than a screen on a smart phone. The customized customer relationship management platform includes a mini mode feature for converting user interactivity to a mini mode for displaying on digital devices including a computer, tablet, laptop computer, smart phone or other digital device for users performing customized user applications at their convenience. The customized customer relationship management platform network includes WI-FI and internet connec-

tivity devices for communicating with various user digital devices. The customized customer relationship management platform network includes digital processors, digital servers, digital computers, digital sensors, digital analyzers, and other digitally controlled devices including wireless digital devices.

[0100] Other features of the Get elements includes Category in Get Integration: Group by for aggregation, Option to Ignore Empty Conditions in Action Button—Get, property in Get—load pick list, and Show Get Integration errors in logs and other features including Sending integration log errors to an additional email, Add group label in Table Add/Edit/View modal windows, Custom labels for Lookup/Table reference, Integration Logs update, and Providing one month of Error log history of one embodiment.

The Flow of Data in a Customized Customer Relationship Management Platform Method and Devices:

[0101] FIG. 3 shows for illustrative purposes only an example of the flow of data in a customized customer relationship management platform method and devices of one embodiment. FIG. 3 shows a customized customer relationship management platform network **300** including a network server. The customized customer relationship management platform **100** digital device is interactively coupled to the customized customer relationship management platform network **300** network server and a platform network attached storage cloud **320** also interactively coupled to the customized customer relationship management platform network **300** network server. The customized customer relationship management platform **100** digital device is used for operating a customized form builder **330** including pdf mapping to convert existing paper forms into responsive digital pdf forms **340** and creating new pdf forms using a dynamic mode and using an auto mode **350**. The customized form builder **330** includes at least one feature including a push data feature for cross platform integration with at least one 3rd party application **360**. The at least one 3rd party application **360** includes Salesforce **380** integration with Salesforce get and Salesforce push wherein Salesforce is a built-in integration feature **370** of the customized form builder **330** of one embodiment.

[0102] FIG. 3 shows using a customized customer relationship management platform network to pass data between a customized customer relationship management platform network attached storage cloud device and a customized customer relationship management platform. Using the customized customer relationship management platform network attached storage cloud device for storing user generated forms and customized integration and other feature data. Using the customized customer relationship management platform for interfacing with users and the customized customer relationship management platform network for using a customized form builder. The customized form builder is used by users for using pdf mapping to convert existing paper forms into responsive digital PDF forms, creating new form using a dynamic mode and using an auto mode. The customized customer relationship management platform is also providing a push data feature for cross platform integration with at least one 3rd party application a user may be already or planning to use. The customized customer relationship management platform provides a single source of form creation and gathering the response data from the responsive form they have created. A user can

integrate Form Builder created forms with a large range of 3rd party applications including Salesforce **380** and others of one embodiment.

[0103] The customized customer relationship management platform includes features that include elements. Those features include Form Builder with Main Operations, Form Elements including Basic Elements, Advanced Elements, Widgets Elements and Element Settings; Form Properties with Settings and Style; Report Builder feature with Report main operations, Report elements including Basic Elements, Advanced Elements and Widgets Elements; and Report settings. A My Submissions feature includes Main operations. Other features include My Forms, My Account, Account Billing, Security, and Sign in, working with the form builder and Form Embed. Salesforce is a built-in integration feature for this 3<sup>rd</sup> party application and includes elements including Salesforce Integration with Salesforce Get and Salesforce Push of one embodiment.

Form Builder Feature:

[0104] FIG. 4 shows for illustrative purposes only an example of a form builder feature of one embodiment. FIG. 4 shows a form builder feature to create new form—options web page **400**. FIG. 4 is showing a blank, template, wizard **410**, Salesforce and from URL. The wizard is used to create a form in 4 easy steps **410** including 1. Form name, 2. Add Elements, 3. On Submit, and 4. Choose Theme. The wizard steps begin with a start button **420**. Pressing the start button **420** takes the user to the form wizard web page **430**. Step 1 enter a form name **440** includes an instruction to give a name to your form, and a description, if needed **450**. The input text box to enter a form name **460** is displayed and the user can enter the form name they assign. An input text box to enter a form description **470** is an optional field this description will appear under the form name as a descriptive introduction. A form direction check box for right to left **480** can be selected for languages that read from right to left of one embodiment.

[0105] FIG. 4 shows how for example to upload an image of a scanned paper form and place the submitted form values above it. This feature is available using basic PDF mapping. A user can create an HTML layout and embed the field values in it. This option is also available using our dynamic PDF mode. Creating a PDF to look like a screenshot of the filled-in form can be done using an Auto PDF feature. Form Builder is a feature that allows a user to easily create powerful forms including elements and sub-features including Field types including Drag and Drop, Responsive, Form wizard, Form Import from URL, Field validations, Unique submissions, Form limits, Mandatory indications, Post submit Redirects, Automatic and custom emails, Element and form Conditional Logic, Save and Resume, Social Autofill, CAPTCHA, Security Seal, Digital Signature, Custom “Thank You” message, PDF Mapping, Multi-page Form, Valid/Error Indicators, Image Masking Backgrounds, Value Rules, Repeated Section, Field Mask, Advanced Values including in a calculated string mode, DropDown with Hierarchy, Workflow Editor Salesforce Form in one click, Payment Integration, Multiple Selection DropDown, Export the form to another account, Custom Translation, Value Rule with REGEX Validation, Bi-Directional Salesforce Integration, Resubmitting entries back to Salesforce and other 3rd party integrations, Form Rules, Salesforce Table Element, Salesforce Visual Composer Targeting, Allow later editing,

Calculated Field, CDN Cache, The Section Element, Collapsible Section, Effected by, Using the Dynamic PDF Mapping, Ignore Mandatory Validation, Address Validation, Auto Submit Your Form, Base a Google map on an address, Version Control feature, PDF Re Generation, Custom mail server, Geo Localization, On Completion Script in Button, Auto PDF, adding a widget to a form, The Salesforce chart, File Upload—Allow Capture, File Upload—Allow Accept, PDF Conditions, Section Repeat with mapped data, PDF Mapping Conditions, PDF Page conditions, PDF Block condition, and Voice Control of one embodiment.

[0106] Other conditions and conditional rules include conditional results for the HREF element: set value and set value from, Set a CSS class to an element by condition, Allow/Disable condition result options for Checkbox and Radio button, filter in condition window, Adding conditions to PDF mapping, Filtering data in Get condition based on an integer, Friendly condition, Friendly condition and brackets added to value rule, Functions added to Field condition results, Is empty condition added in Numeric and Price, condition result options: Read only and Read Write, Opposite rule in condition, Optimization for loading times, loading times, CDN cache, conditional logic engine, PDF conditions—if block, Reset radio option in condition results, Show/Hide section tabs via condition, Supporting Boolean conditions, and Supporting special characters in condition of one embodiment.

[0107] Other features include feature: voice control, features to Advanced Tools, features to Google Sheets Export, feature for multi-page forms: Progress indicator, feature: Auto Submit, Version Control feature, Lock elements in form builder, Hotkeys in form builder for Mac, and Showing validation indicators on form builder canvas of one embodiment.

[0108] FormTitan offers all the elements to create a landing page or online form: Input elements, graphic elements, special elements and widgets. They are all located in the left “Elements” panel and divided into 3 categories: Basic, Advanced and Widgets. Basic elements include Section, Image, Button, Line, Label, Paragraph, Href, Heading, Bullet, Textbox, Numeric, Textarea, Email, URL, File Upload, Dropdown, Radio, and Checkbox. Advanced elements include Address, Full Name, Date & Time, Date, Time, Phone, Price, Star Rating, and Likert. Widgets are elements that include YouTube, Vimeo, Seal, Privacy, Signature, Social Filler, ImageBlock, ImageSlider, HTML, Google Map, Calculated, Hidden, and Page Break of one embodiment.

#### Form Builder Uploading a User's Existing Pdf Form:

[0109] FIG. 5 shows for illustrative purposes only an example of form builder uploading a user's existing pdf form of one embodiment. FIG. 5 shows the form builder uploading a user's existing pdf form 500. A listing of user existing pdf forms is displayed for selection of an existing pdf form. The user selects user's existing pdf form #1.pdf 530 from the listing. The uploaded user's existing pdf form 525 is displayed on a form builder web page 510 that includes in a header section a mapping field web page 520. The user can then open an elements 550 tab to make element selections to integrate into the user's existing pdf form 525 of one embodiment.

[0110] FIG. 5 shows how to easily convert a paper form into a responsive form and automatically output online form

entries into the paper form. 1. SCAN: Scan an existing paper form and save it as an image or non-responsive pdf. This scanned image will be used in step 3. 2. BUILD: Build an online version of the scanned form using FormTitan form builder. 3. MAP: Upload the scanned form and map online form fields to it. 4. USE: Start using the online form. Upon data submission a PDF version will be generated. Responsive Form Create a responsive online form to collect data via P.C., tablets and mobile phones. Password Protected Protect an offline form by sealing it with a password. Digital Signatures Collect signatures using an online form. Layouting & Kerning Map the online form to the corresponding paper form. Easy to use Simple WYSIWYG online form builder with drag & drop.

[0111] Many financial, insurance and legal services still rely on paper forms, which have to be downloaded, printed, manually filled in and sent back via fax. This is an inefficient, inconvenient, time consuming process which leads to negative customer impression and consequently also to low conversion. FormTitan allows a user to bypass these shortcomings by collecting the data using an online form, and still receiving it as a PDF in the original layout.

[0112] FormTitan provides help with tutorials and specific instruction including GETTING STARTED, Usage Guidelines, Watch a tutorial, Forgot Password, How do I create a form? FORM BUILDER, form builder layout, Apply a theme on my form, Change elements style, Types of forms I can build. MY SUBMISSIONS, Submissions Dashboard, Open form Submissions, Emails sent upon submission, Submissions Page Layout. MY ACCOUNT, My Account Profile, account profile images, Add a collaborator under the user account, Change account password. BILLING, My billing information, Billing Transaction Receipt. SECURITY, CAPTCHA, makes sure no one enters a user account. Integration into 3<sup>rd</sup> party applications including Salesforce, Salesforce, Salesforce get integration, How can I track my Salesforce API calls?, Salesforce Push Integration Example #1, Salesforce Get Integration Example #2. A Password Mode Textbox is a feature to use in the protection of a user's privacy of one embodiment.

#### Drag and Drop the Form Builder Elements Over Info Locations on an Existing User Scanned Paper Form:

[0113] FIG. 6 shows for illustrative purposes only an example of drag and drop the form builder elements over info locations on an existing user scanned paper form of one embodiment. FIG. 6 shows drag and drop the form builder elements over info location on a user existing pdf 600. The mapping field web page 520 is displayed from a FormTitan web site 610. The elements 550 include a drop down elements menu including a label selection. A user drags the label element to the user's existing pdf form 525. A drag and drop element 650 for example a label element 660 is position on a user's existing pdf form and is entered in the mapping fields web page 620 of one embodiment.

[0114] FIG. 6 shows how to build an online version of a scanned form using the form builder. PDF Mapping allows a user to 1—upload a PDF and use it as the background to an online PDF form. 2—Select the form fields from the Element Dropdown (one by one) and place them where their value appears.

[0115] According to “TechTerms” The drag & drop feature “involves moving the cursor over an object, selecting it, and moving it to a new location”. FormTitan form builder is

a WYSIWYG tool and it is easy to use. It allows a user to create landing pages and forms intuitively by dragging elements from the “Elements” panel on the left, and dropping them in the canvas. The elements may be placed anywhere since the layout is completely flexible. DropDown with Hierarchy. FormTitan provides a “Dropdown” element a user can add to the user form to. With this element the form filler can input data in a form by choosing from a list of options. FormTitan, however, provides two types of Drop-Down: a regular Dropdown called “simple”, and a dropdown with a few levels called “Tree” (which is in fact a Dropdown with Hierarchy). Creating a Dropdown with Hierarchy instructions: 1—Enter the form builder and create a new form or open the form to add it to. Since the “simple” Dropdown has only one level, it can be changed to a “tree” by adding Hierarchy to it. The following example shows how to create a “Tree” Dropdown from scratch. 2—Drag a Dropdown. 3—Once selected go to its properties and change its label text. “Properties Panel”>“Element” tab>Settings>“Basic” category>Label. In the example created the dropdown is called “School Kids”. 4—Now change the Mode to “Tree” “Properties Panel”>“Element” tab>Settings>Mode Radio button>Tree. 5—Press on the “Configure” button. 6—A window will open containing all of the Dropdown’s default options. Displaying 3 rows. Each row is an item in the highest level of the tree. (Parent).

[0116] Each row has its own icons to do the following: Edit—change each item’s name with the “Edit” Icon. Delete—remove an item by pressing on “Delete”. Add—add new options under a specific row (Child) by pressing on the “Add” button (+) in that row. In addition there is an “Add object” button to add more items to the parent level. An example is configured a tree with 3 levels: The parent level, which contains schools: Elementary school, Middle school and High school. Under the “Elementary” school option is added another level (child) with 2 options: ‘Class 1’ and ‘Class 2’ and under “Class 2” is added another level with 3 kids names. 7—Press “Apply” and save the form. 8—Publish the form to see for how the Dropdown will appear to users. The items on the list will be displayed with an indentation to show the Hierarchy. 9—in addition to this, once data is entered in this dropdown and submitted a viewer will be able to see the data displayed in a Hierarchy, in a submissions page. It will be displayed in this manner in an email, submissions page and exported data of one embodiment.

#### A Form Builder Pdf Mapping Menu:

[0117] FIG. 7 shows for illustrative purposes only an example of a form builder pdf mapping menu of one embodiment. FIG. 7 shows a form builder pdf mapping menu 700 on a form builder pdf web page 705. The pdf mapping menu 700 includes a form builder pdf grid layout page 720 that includes a horizontal ruler 710 and a vertical ruler 730 of one embodiment.

[0118] FIG. 7 shows a form builder pdf mapping menu including a file dropdown, save, desktop dropdown, preview, and publish, tips and help. Using the Dynamic PDF Mapping starts with using the advanced PDF Mode. Paper forms have been around for ages. Today many forms have gone online, however there are still services which rely on paper forms (like law offices, insurance companies, etc.). The problem with paper forms is that they slow things down, they require download, printing, filling in manually, and

sending by fax. This becomes a hassle since many people don’t have a printer or a fax. In addition, filling in forms by hand can get tiresome when required to fill in the same fields over and over again (like name or signature).

[0119] Furthermore, Companies that need to digitally store the data that was collected with their paper forms now have to deal with a heavier work load that includes scanning the forms, typing in the data etc. This is exactly why FormTitan developed this feature: PDF mapping. A PDF (Paper Form) Mapping feature allows a user to collect the user data using a responsive online form, and then have a PDF document generated based on the entry, with the data in it. This PDF can be sent to Form Builder creator user, the form owner, to the assigned form fillers and it can be pushed to for example Salesforce. This is a win-win situation where the data is entered and stored digitally as well as generated into a PDF document in the traditional old way.

[0120] FormTitan provides 3 types of PDF mapping: 1—A basic PDF mapping where a user uploads a background, and places form elements above it for example using basic PDF Mapping. 2—A Dynamic PDF where the user creates the layout using an editor and HTML code. 3—An auto PDF where the system creates the PDF by capturing an image of the filled in form (sort of like a screenshot) of one embodiment.

[0121] Auto PDF:

[0122] 1—Create a form in the form builder. 2—Go to ‘Properties’ panel>‘Form’ tab>‘Settings’ option>‘PDF Mapping’ category>Auto PDF. 3—Press on the ‘Map Fields’ button. 4—A small modal window will open to configure the settings:—Allow auto PDF mapping: this checkbox will activate the auto PDF. Auto Fit: this will make the screenshot fit the A4 PDF size. Encrypt PDF: A user can have the PDF attached to the user emails encrypted, this means that it will only open with a password. With this dropdown a user can choose which field will be used as password. Message with password for PDF: In this text area a user can configure a message that will appear when someone tries to open an encrypted PDF. Custom file name: A user can enter a name for the PDF being created.

[0123] This will be the file name of the PDF attached to the email. Send to owner/Send to additional/Send to form emails: Checkboxes that determine where this paper form will be sent to. Allow Condition: if a user decides to turn on this Checkbox the user will be allowed to add a condition by which this PDF will be generated. Only if the condition terms are met then the PDF will be generated. Process upon payment only: turning on this checkbox will make sure that the PDF will be generated only if the payment is executed. Since this PDF is a screen capture of the form, there will be no need to create a layout in the editor (as is done in the other 2 PDF options). 5—Save the form and it is ready to use. 6—Now a user can test the form: Publish it as URL, enter data in it and submit the form. Other features are used in the processing of payments including Override of product name in Payment Integration platforms, Payment Integration Emails, Payment integration including single and recurring payments with various online payment platforms of one embodiment.

[0124] Other mapping features include Conditional Logic in the PDF Mapping, Mapping only reference fields in a user created Table, PDF Mapping: Auto PDF, PDF Mapping Additions, PDF Mapping file name and barcode, Remove auto margin in Dynamic PDF Mapping, Vertical Align

Added to Elements in PDF Mapping, Adding conditions to PDF mapping, Push mapping Condition, Conditional mapping in Repeated and Files, Else added to the Conditional mapping, My submissions filter by PDF mapping, pdf mapping—categories, PDF mapping generated after custom Push, Submission ID in mapping of nested child, and Using the Short State name in Address mapping of one embodiment.

#### Advanced PDF Mapping Dynamic Mode:

[0125] FIGS. 8-19 show features of the Advanced PDF mapping Dynamic mode including elements.

#### Form Builder Pdf Mapping Dynamic Mode Basic Elements:

[0126] FIG. 8 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode basic elements of one embodiment. FIG. 8 shows form builder pdf mapping dynamic mode basic elements 800 on a form builder pdf mapping dynamic mode basic elements menu and grid layout web page 810. An elements menu 815 includes basic elements 820. The basic elements 820 menu includes section 830, image 840, button 850, line 860, text 870, input 880, and multiple input 890 of one embodiment.

[0127] FIG. 8 shows when adding PDF Mapping to a form the standard editor opens, allowing a user to easily create a PDF by uploading a user document, and then placing form fields above it. It's really simple to do. However, for users who do not want to create a PDF this way a Dynamic Mode is available. In Dynamic Mode the editor will change, allowing a user to insert HTML code, play with styling and even use a table element for displaying repeated section items dynamically. For example: 1—Create a new blank form; 2—Move submit button down—so it is located after the section; 3—Drag a few elements on to the canvas. Other Dynamic mode features include Dynamic PDF Mode, Dynamic PDF Background image, Page order in Dynamic PDF, Remove auto margin in Dynamic PDF Mapping, and Setting a Dynamic reply to of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode Basic Text Element:

[0128] FIG. 9 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode basic text element of one embodiment. FIG. 9 shows for example a form builder pdf mapping dynamic mode basic text element 900 for selection of a sub-element to be positioned on the form builder pdf mapping dynamic mode basic elements menu and grid layout web page 810. A user makes a selection from the basic elements 820 in this example text 870. The text 870 includes a sub-menu that includes Label 940, Paragraph 950, HREF 960, Heading 970, and Bullet 980. In this example a user makes a label selection 945 to position label text box as an anchor text 947 on the grid layout of one embodiment.

[0129] FIG. 9 shows a Textbox→change its label to “Parent Name” Drag a Section, and make it wider to fit the 3 following elements: 1. textbox→change the label to “Child name”, 2.—numeric→change the label to “Child age”, and 3.—textbox→change the label to “favorite color” of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode Basic Input Element:

[0130] FIG. 10 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode basic input element of one embodiment. FIG. 10 shows a form builder pdf mapping dynamic mode basic input element 1000 on the form builder pdf mapping dynamic mode basic elements menu and grid layout web page 810 basic elements 820 menu. The input 880 menu selection by the user displays a sub-menu that includes selections that include a Textbox 1040, Numeric 1050, TextArea 1060, Email 1070, and URL 1080. The user can for example make selections and place them on to the grid layout including a positioned textbox 1045, positioned numeric box 1055, positioned textarea box 1065 and positioned email box 1075 of one embodiment.

[0131] Other elements include label to Section element, default state of the element to Disabled, Custom subject in the Email element, Hover and Selected effects in elements, Infinite scroll in the Section element, element for Salesforce: Chart, element: HTML Editor, target options in the HREF element, Positioning elements with Intelligent Drop, Set Value From—added to radio and dropdown elements, Size per page in pagebreak element, and Styles for buttons in Table element including alignment of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode Basic Multiple Input:

[0132] FIG. 11 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode basic multiple input of one embodiment. FIG. 11 shows a form builder pdf mapping dynamic mode basic multiple input 1100 displayed on the form builder pdf mapping dynamic mode basic elements menu and grid layout web page 810. The user can select from the basic elements 820 menu the multiple input 890. The multiple input 890 selection displays a sub-menu that includes a dropdown 1140, radio 1150, and checkbox 1160 selection. A user can for example place a positioned dropdown selection box 1145 on the grid layout. The user can for example place a positioned radiobutton including first choice second choice third choice and other 1155 on the grid layout. The user can for example place a positioned checkbox including first choice second choice third choice and other 1165. FIG. 10 shows 4—Select the section and go to: “Properties” panel>“Element” tab>“Settings” option>“Basic” category of one embodiment. FIG. 11 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode basic multiple input of one embodiment. FIG. 11 shows turn on the “Repeated” option. This will allow the form filler to add button and enter multiple items instead of just one. Save the form of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode Contact:

[0133] FIG. 12 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode contact of one embodiment. FIG. 12 shows a form builder pdf mapping dynamic mode advanced contact 1200 selection for the user to incorporate into their custom pdf form. The form builder pdf mapping dynamic mode an advanced elements menu and grid layout web page 810 includes advanced 1220 elements for a user to create a contact 1230 section. The contact 1230 section sub-menu includes address 1240, full-name 1250 and phone 1260 selections of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode Date and Time:

[0134] FIG. 13 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode date and time of one embodiment. FIG. 13 shows a form builder pdf mapping dynamic mode advanced date and time **1300** selection for the user to incorporate into their custom pdf form. The form builder pdf mapping dynamic mode an advanced elements menu and grid layout web page **810** includes advanced **1220** elements for a user to create a date & time **1330** section. The date & time **1330** selection includes sub-menu selections including datetime **1340**, date **1350**, and time **1360** of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode Range:

[0135] FIG. 14 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode range of one embodiment. FIG. 14 shows form builder pdf mapping dynamic mode advanced range **1400** section on the form builder pdf mapping dynamic mode advanced elements menu and grid layout web page **810**. The range **1420** selection includes a sub-menu including a starring **1430** and slider **1440** elements of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode Video:

[0136] FIG. 15 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode video of one embodiment. FIG. 15 shows the form builder pdf mapping dynamic mode widgets elements menu and grid layout web page **810**. The form builder pdf mapping dynamic mode widgets elements menu includes a video **1500** section. The widgets **1520** video **1530** selection includes a sub-menu including YouTube **1540** and Vimeo **1550** of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode Privacy:

[0137] FIG. 16 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode privacy of one embodiment. FIG. 16 shows the form builder pdf mapping dynamic mode widgets elements menu and grid layout web page **810**. The form builder pdf mapping dynamic mode widgets elements menu includes a form builder pdf mapping dynamic mode privacy **1600** section. The elements **550** include in the widgets **1520** a privacy **1630** selection. The privacy **1630** selection includes in a sub-menu Seal **1640**, Privacy **1650**, Signature **1660**, and Social Filler **1670** selections of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode Image Group:

[0138] FIG. 17 shows for illustrative purposes only an example of form builder pdf mapping dynamic mode image group of one embodiment. FIG. 17 shows a form builder pdf mapping dynamic mode image group **1700**. In the elements **550** widgets **1520** selections is an image group **1730** that includes in a sub-menu an Imageblock **1740** and Imageslider **1750** for selection by a user of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode Settings:

[0139] FIG. 18 shows a block diagram of an overview of form builder pdf mapping dynamic mode settings of one embodiment. FIG. 18 shows form builder pdf mapping dynamic mode settings **1800** elements **550**. Under a settings (fld1) **1820** is a basic (button) **1830** selection that includes

user selections including button text: **1840**, user tip: **1850**, button type **1860**, and on completion script **1870** selections of one embodiment.

[0140] Other features are used for pdf mapping and handling including Adding pages to dynamic PDF, Charts in Auto PDF, Conditional Logic in the PDF Mapping, Generate pdf for all entries in my submission, Maximizing the PDF preview window, Merge attachment and files from Salesforce into PDF, My submissions filter by PDF mapping, page break inside new pdf, PDF Auto preview, PDF generated upon payment, PDF generated upon payment, PDF Interactive mode, PDF kerning works with Interactive mode, pdf mapping—categories, PDF Mapping Additions, PDF Mapping file name and barcode, PDF mapping generated after custom Push, PDF Mapping: Auto PDF, PDF sent to email upon payment, PDF supports letter format, Private content in PDF—Do not include in preview, Processing message for download pdf/auto pdf/file upload in my submissions, Push Signature to Salesforce as attachment without PDF Mapping, Salesforce action Override mode for auto pdf, Vertical Align Added to Elements in PDF Mapping, and View manual/auto PDF information of one embodiment.

#### Form Builder Pdf Mapping Dynamic Mode:

[0141] FIG. 19 shows a block diagram of an overview of form builder pdf mapping dynamic mode of one embodiment. FIG. 19 shows a form builder pdf mapping dynamic mode **1900** on a FormTitan webpage **1910** that includes an elements selection menu **1920** including basic **1930**, advanced **1220**, widgets **1520**, and Salesforce **1960** of one embodiment.

[0142] 5—Now lets add the PDF Mapping. Go to: “Properties” panel>“Form” tab>“Settings” option>“Basic” category>PDF Mapping.

6—Press on the “Map fields” button and a window will open—containing the standard PDF mapping editor. Select the Dynamic mode from the upper toggle.

7—The editor will now switch to the Dynamic PDF editor

8—Now a user can enter content in the editor. A user can make this a multi-page PDF by adding pages. Simply press on the ‘+Page’ in the toolbar and a new page will open in a separate tab.

[0143] A user can also add pages by inserting a Page break in the PDF editor from the ‘Insert’ option, and it will divide the PDF into pages. However, this way a user will not see the pages in different tabs, but rather a horizontal separation line that will divide the content in the editor. 9—Inserting HTML. Go to Menu>“View”>“Source code” and paste the user HTML code in the window. 10—Apply Styles. A user can enter text and style it: select font, size, color, alignment etc. 11—Adding a table with repeated section data. Go to Menu>“Insert”>“Table” and create 1 row and 3 columns. The table will now appear in the editor. It will only have one row, and a user will need to just map it to the 3 section fields. More rows will dynamically be added based on the number of rows the form filler added in the repeated section. In the left column—Click inside the column and Enter text: “Child Name:”—Open the “All element” dropdown and Choose the Child name field it will look like this: In the center column—Click inside the column and Enter text: “Child Age:”—Open the “All element” dropdown and Choose the Child Age field it will look like this: In the right column—Click inside the column and Enter text: “Favorite color.”—

[0144] Open the “All element” dropdown and Choose the Favorite color field it will look like this: 12—Press on “Settings” in the upper toolbar and configure the PDF properties and set the following:—Open the ‘General’ category and turn on the “Preview before submit” checkbox.—Open ‘Buttons & labels’ category and enter a different text for the finish button, Cancel button and preview window title (they already have default values, so this is optional).—Open the ‘Emails’ category and turn on the “Send to owner” checkbox. This will make sure this PDF is sent to the form owner once the form is submitted. 13—Press on “Apply”. 14—The dynamic PDF being created will be generated in an A4 size document.

[0145] The PDF will have default margins on all sides, if a user wants to remove them the user can turn on the ‘Remove auto margin’ checkbox. If a user wants to add customized margins, the user can do this in the Html Source of the PDF by navigating to View>Source Code. 15—Save the form. 16—Test the form: Publish it as URL and enter a parent name and details for 3 children, (press twice on “Add” to get three lines). Press on submit—and a preview of the generated PDF will appear. If a user presses on “Finish” the entry will be submitted and the PDF will be attached to the email sent to the form owner.

[0146] Dynamic PDF Mode is designed for users who are using or intend on using the PDF Mapping. Dynamic PDF Mode is a pdf Mapping feature that allows a user to use HTML code to create a user PDF, to play with styles and add a dynamic table that is populated from a repeated section in the form.

[0147] The dynamic PDF mode provides an editor for creating a PDF. Enter the user text or insert existing user HTML code. The user can then use the editor options to set the styles and the user can now insert a table to this PDF, and map it to a repeated section.

[0148] Page Break Inside Dynamic PDF, the dynamic PDF has its own editor that allows a user to create a PDF using HTML, styles etc. including creating a multi-page PDF. A multi-page PDF can be created in both the regular PDF mode and the Dynamic mode. A user can insert a page break to a PDF in the editor. Each time a Page break is inserted it will display horizontal lines where the break should be.

[0149] Adding a Widget to a Form.

[0150] A user can add a widget/chart or any kind of an external logic into a form. An external widget can interact with form fields, whether the form is filled manually by users or populated from Salesforce. The steps are: 1—add a HTML widget into the form. 2—use the window.parent. ftGetValueByID('FIELD ID')) function. The following example is to add a widget to a form for example a gauge widget on the form that will respond to a FormTitan slider field. 1—Create a new form. 2—Add a slider element. 3—Drag an HTML element to the form canvas. 4—Paste the widget code in the HTML element: <script type="text/javascript">, </script>. 5—Replace the FIELD ID with the actual field ID of the selected slider or any other field on the form. 6—The widget appears in the HTML element, on the form canvas. Resize the HTML element and move it to the right of the Slider field. 7—Select the slider and change its Max number to 200. 8—Select the HTML element and remove its border go to: “Properties” panel>‘Element’ tab>‘Style’ option>‘Border’ category>Border style=None. 9—Save the form and publish it as URL.

[0151] Digital Signature

[0152] The “Digital Signature” is an element a user can add to an online form when the user needs a form filler to sign for identification and Consent, for example: in a legal document, agreements, order forms etc. Following are steps to add a digital signature to a form: 1—Enter the form in the form builder. 2—Go to the “Elements” panel>“Widgets” category>“Privacy” sub category. 3—Drag the “Signature” element onto the form canvas. A user can change its style in the “Properties” panel.

[0153] Auto Submit Your Form.

[0154] The form owner may not want the form filler to enter data in the form. The form owner wants to auto populate the fields and have the form automatically submitted. To do this the form user can select an “Auto Submit” feature. Following are the steps to make the form auto submit: 1—Enter the form in the form builder. Since this form is intended for auto submit, a form owner should make sure the data is populated in the form automatically when it loads. The form owner can use Params in the URL for this, Salesforce Get integration and conditions to set values in fields based on input. 2—Go to “Properties” panel>Form tab>Settings option>On submit category. 3—Turn on the “Allow Auto Submit” checkbox an input box will appear below it and the form owner will be required to enter a string of 16 alpha numeric characters, for example: abc234fr48yttr32. 4—Save the form. 5—Publish the form. 6—Add the following to the form URL: ?FTAutoSubmit=abc234fr48yttr32 and the 16 alpha numeric characters will append to the Browser address bar. 7 Press on “Enter” and the form is loaded, populated and auto submitted.

Integration Form Builder Pdf Mapping Salesforce:

[0155] FIG. 20 shows a block diagram of an overview of integration form builder pdf mapping Salesforce of one embodiment. FIG. 20 shows an integration form builder pdf mapping—sales force 2000 Salesforce MVP webpage 2010. The integration form builder pdf mapping—sales force 2000 Salesforce MVP webpage 2010 is used for Salesforce integration 2020 into a user's forms. Other features for Salesforce integration 2020 and functionality include Mapping the Salesforce Checkbox to a Radio button/Dropdown/Multi dropdown, Reset button in Salesforce Push/Get mapping, Salesforce Action button—Now supports Get, Go to Next or Previous Page in Salesforce Action Button, Salesforce Action Button in a multi-page form, Salesforce Action Button per repeated item line, Mapping fields from levels in your Salesforce Table, Mapping the Salesforce Checkbox to a Radio button/Dropdown/Multi dropdown, Mapping the Salesforce Country/State picklists, Push Signature to Salesforce as attachment without PDF Mapping, Controlling which fields may not be editable or creatable in Salesforce mapping, Reset button in Salesforce Push/Get mapping, Salesforce Push mapping to not override in a specific field, Supported boolean in Salesforce table mapping, font—Salesforce Sans, operators for Salesforce Get/Push, Skip Message in Salesforce Get Action, Adding an HREF to The Salesforce Table, Salesforce Table: Mandatory in Edit and Add, Alternative Find In Salesforce Get, Caching object list from Salesforce, Create Upon Payment Label in Salesforce integration, Displaying data from Salesforce in an online map, Document generation (single/bulk) directly from Salesforce, Dynamically Limiting Rows in Salesforce Table

with -1, Executing actions after Delete in Salesforce Table, Executing actions after View/Edit/Add in Salesforce Table, Export added to Salesforce Table, Fields Order in Salesforce Form, Go to Next or Previous Page in Salesforce Action Button, Hide Salesforce Table rows footer, Ignore Assignment Rules in Salesforce, Inline editing in the Salesforce Table, Limit num of rows in a Salesforce table based on parameters, Load custom list from Salesforce, Merge attachment and files from Salesforce into PDF, category in Salesforce Get: Translation, publish option: Shadow form (Salesforce web-to-lead), Push Edit later URL to Salesforce, Pushing labels to Salesforce, Replace BR with CRLF in Salesforce Push, Replacing Empty with Null in Salesforce Push, Salesforce Action button—Now supports Get, Salesforce Action Button in a multi-page form, Salesforce Action Button per repeated item line, Salesforce action Override mode for auto pdf, Salesforce Get on load based on a hidden field, Salesforce Table Edit/View/Delete links with Icons, Salesforce Table Max rows per level, Salesforce Table Modal Windows Style, Salesforce translation support for Global Picklist Value Sets, Save as draft working seamlessly with Salesforce integration, Store link to Allow later editing in Salesforce, Submission id in Salesforce push condition upon payment, Supporting Checkbox for multi-picklist in Salesforce, Supporting Child Objects in Salesforce Push Payment, TIME field in Salesforce table—Update and Add, Using a Hyperlink From Salesforce in Your Table, Extended support when mapping Get with Salesforce lookup fields, Refresh Salesforce object list from mapping, element for Salesforce: Chart, Dynamically Limiting Rows in Salesforce Table with -1, Submission id in Salesforce push condition upon payment, Pushing labels to Salesforce, Adding an HREF to The Salesforce Table, Alternative Find In Salesforce Get, Caching object list from Salesforce, category in Salesforce Get: Translation, Create Upon Payment Label in Salesforce integration, Creation Option Added to Salesforce Push—if Multiple Found, Dates format in Salesforce Table View/Edit/New modal windows, Displaying data from Salesforce in an online map, Extended support when mapping Get with Salesforce lookup fields, Fields Order in Salesforce Form, font—Salesforce Sans, Go to Next or Previous Page in Salesforce Action Button, Hide Salesforce Table rows footer, Ignore Assignment Rules in Salesforce, Load custom list from Salesforce, Mapping fields from levels in your Salesforce Table, Mapping the Salesforce Checkbox to a Radio button/Dropdown/Multi dropdown, Mapping the Salesforce Country/State picklists, Move to a specific page using the Salesforce Action button, operators for Salesforce Get/Push, Order of the fields in the Edit Add Salesforce Table windows, publish option: Shadow form (Salesforce web-to-lead), Push Edit later URL to Salesforce, Push logs now reflecting Salesforce Table actions, Replace BR with CRLF in Salesforce Push, Replacing Empty with Null in Salesforce Push, Reset button in Salesforce Push/Get mapping, Salesforce Action button—Now supports Get, Salesforce Action Button in a multi-page form, Salesforce Action Button per repeated item line, Salesforce Get on load based on a hidden field, Salesforce Push mapping to not override in a specific field, Salesforce Table Edit/View/Delete links with Icons, Salesforce Table Max rows per level, Salesforce Table Modal Windows Style, Salesforce Table: Mandatory in Edit and Add, Salesforce translation support for Global Picklist Value Sets, Save as draft working seamlessly with Salesforce integration, Skip Message in

Salesforce Get Action, Store link to Allow later editing in Salesforce, Supported boolean in Salesforce table mapping, Supporting Checkbox for multi-picklist in Salesforce, Supporting Child Objects in Salesforce Push Payment, TIME field in Salesforce table—Update and Add, Using a Hyperlink From Salesforce in Your Table, Controlling which fields may not be editable or creatable in Salesforce mapping, Refresh Salesforce object list from mapping, and Salesforce Table with row limit of one embodiment.

[0156] **Salesforce Get Category—First Steps**, Once a user presses on the “Map Fields” button the “Get from Salesforce” window will open. This is where form builder selected integration will be configured and displayed. “Get” is a very general term that refers to drawing data from an integrated application platform for example one or more Salesforce objects. Each Get action performed on a Salesforce object will be represented in one integration line, so if a user wants multiple objects being read the user will need to create an integration line for each. The first time a user opens this window—the form will have no lines and a user will need to press on the “Add object” button to create the first integration line.

[0157] Once an integration line is created it will be added to this window and the user will be able to edit it using four icons on the right side of the line: edit the line, duplicate the line, delete it or add an integration with a child object. First step—Authentication: 1—Authenticate with the user’s Salesforce account. Press on the “Authenticate” button and login to the user’s Salesforce account. If the authentication will succeed a green V icon will appear. 2—Press on an “Add object” button to specify the object the user wants to connect to in the user’s Salesforce account. This will open a window where the user will be able to start configuring the Get integration. The integration window will open containing 2 categories (out of 7) in an accordion structure: Once the user has chosen the object the rest of the categories will appear and the user will be able to go on with setting the Get integration.

[0158] **Salesforce Get Category—**

[0159] Connection is the first category in the accordion. This category contains two fields: 1—“Connection” label—specifies the Salesforce account currently connected to. 2—“Use different Salesforce connection” checkbox—this is designed for changing the current connection to a different Salesforce account; however a user may not use a different connector without a written consent from FormTitan.

[0160] **Salesforce Get Category—Map Fields,**

[0161] Map Fields is the Seventh category in the accordion and the core of the integration line: this is where a user can map form fields to Salesforce object fields. Map Salesforce fields includes select an object, conditions, rule conditions, sort order, map fields, and messages. This category usually contains one field: 1—‘Map fields’ button—this button will open a modal window where the user will be required to map form fields to the fields in Salesforce. Choose a form field, open the Salesforce field dropdown and choose the field to map it to. This button will only appear if there are more than four fields in the form. If there are four or less fields the mapping lines will already appear instead of opening in the modal window. A filter is used to correlate a form field to a corresponding Salesforce field. A user can use the filter to look for a specific field, or use the “show” drop down to view “All”/“Mapped fields”/“Not mapped fields”.

[0162] Salesforce Get Category—

[0163] Condition is the third category; a user will need to fill it in right after ‘Salesforce object settings’ category. This category contains four fields: 1—‘Set Condition’ button—this button will open a modal window where a user will need to set a condition. If a “get” integration line is not executed on form load (which is flagged in the previous category) a user will be required to enter a condition for drawing the data. A user will need to choose: a Salesforce field, an operator and a form field (or other value). For example: Account Name Equals Account, a user can press on the “Add new condition” button to add rules or on the minus button to remove rules.

[0164] Pressing on the “Show friendly condition” link will let a user see the condition is a clearer format. Turning on the “Remove empty conditions” checkbox—This option makes sure that if a field in a FormTitan form is empty/null it will not add the condition to the query. 2—‘If multiple matches found take ID from’ radio button—This field will allow a user to decide what happens when multiple matches are found. A user can choose to use the first result, the last result, use all of the results, or choose “Skip” to pass over mapping altogether. 3—“Limit” numeric textbox—if user chooses “All” in the previous radio button the user can narrow down the results by drawing only part of them. 4—“ID of section” dropdown—This dropdown contains a list of all the sections in the form. It should be used when the user wants to display many results in a repeated section.

[0165] Since there are sometimes many section elements in one form, and even sections within sections, it may be hard and even impossible at times to cause a specific section to be repeated automatically. This is where this dropdown is used; allowing the user to choose which section will be repeated to display many items. 5—“If not found—run” checkbox—This option is designed for when an integration line returns empty—with no matches found and the user wants the search to continue by running a different integration line. For example: when a user asks for the form filler’s email to find out if the form filler is a registered contact or lead. A user can look for this email in the contact object first, and if it is not found three, run the next integration line, and look for it in “Leads” object.

[0166] Salesforce Get Category—

[0167] Rule Condition is the fourth category, there is nothing required in this category. A user can use it to create a condition, specifying when this “Get” integration will work. This category contains one field: 1—‘Rule Condition’ button—this button will open a modal window allowing a user to set a condition. For example: Textbox Contains “John”. A user can create a more complex rule by pressing on the “Add new condition” button, using brackets and putting “And”/Or between the conditions.

[0168] The FormTitan integration feature includes coordination of time zones between the form builder time zone and the time zone wherein the 3<sup>rd</sup> party application is located. In this example to reflect in the form builder created forms the Salesforce time zone to prevent any discrepancy the form builder date time element in the form is mapped to a Date/time field in the Salesforce account.

[0169] Adding a Chart Based on Salesforce Data to the Form.

[0170] The Chart element under the “Salesforce” category in the elements panel allows a user to display information from Salesforce in a graphic way, in real time. A user can add

a bar chart that shows the number of accounts according to each account type. Following are steps: 1—drag a chart element. 2—A chart settings window will automatically open and choose the chart type. 3—Choose Bar and press on “Next”. 4—Press on the “configure Salesforce integration”—Select an object: Account. 5—Press “Next”—Set the chart as follows: Aggregate By: Count, Count By: Account ID, Chart period: without, Label: Account type, Group By: Account type—Press ‘apply’. 6—Select the chart on the form canvas and make it bigger. While the chart is selected a user can press on the “Map fields” button and change its settings. To add other types of charts a user can do the same exact process, only choose a different chart type. 7—Test the form: Publish it as URL.

Integration Form Builder File Upload—Allow Accept:

[0171] FIG. 21 shows a block diagram of an overview of integration form builder file upload—allow accept of one embodiment. FIG. 21 shows an integration form builder file upload—allow accept 2100 element 2110 for a settings (fld2) 2120. The advanced 2125 element includes an allow accept 2130 of a file-upload by a user of one embodiment.

[0172] File Upload—Allow Accept, is used to allow the user to filter the user directory of files to choose from for an upload. Rather than bringing up all of the files in the user’s personal computer directory to make a choice, the File Upload—Allow Accept element narrows the listing on the browser window opens it only list the type of files selected by the user. Only files relevant to the selected file type are shown.

[0173] For example: If a user only wants a form filler to upload an image (a GIF, PNG or JPG), there would be no point in displaying irrelevant file types like Doc, CSV, PDF, HTML etc. Following are the steps to use the ‘Allow Accept’ option: 1—Select the file upload element on canvas. 2—Go to ‘Properties’ panel>‘Element’ tab>‘Advanced’ category Turn on the “Allow accept” checkbox. 3—Specify the file types the user wants the form filler to see. Make sure to start each file type with a dot and separated multiple file types with a comma—for example: jpg, png. 4—Save the form and test it—Publish the form as URL,—Click on the file upload field,—The browse window will then open—and displays either JPG or PNG file types.

[0174] File Upload—

[0175] Allow Capture is an element for a user uploading files and attaching them to the form created using form builder. Form builder includes a file upload element to allow a user to upload files using a ‘Capture’ property that extends this function of the file upload element. The capture property allows a user to capture an image, an audio or video right then and there, and attach it to a form. File Upload—Allow Capture The file upload element allows form fillers to upload files and attach them to the form. The ‘Capture’ property extends this functionality, because it allows the form filler to capture an image, an audio or video right then and there, and attach it to the user form.

[0176] When the form filler clicks on the File upload field—this immediately opens the relevant application and allows the form filler to capture the file. Following are steps to use the ‘Allow capture’ option. 1—Select the file upload element on canvas. 2—Go to ‘Properties’ panel>‘Element’ tab>‘Basic’ category. Turn on the ‘Include in email’ checkbox. 3—Go to ‘Properties’ panel>‘Element’ tab>‘Advanced’ category, Turn on the “Allow capture”

checkbox. 4—The display will show three options:—Camera: used to capture images,—Microphone: used to capture audio and—Camcorder: used to capture a video. 5—Choose one of these options—in this example: Camera. 6—Turn on the “Allow accept” checkbox and enter a suitable value: If the choice is Camera→Type: image/\*, If the choice is Microphone→Type: audio/\*, If the choice is Camcorder→Type: video/\*. For example the choice could be “Camera” in the previous step—the user can type: image/\*. 7—Save the form and test it,—Publish the form as URL,—send it to the user directly and open it in a Mobile phone browser.—Click on the File upload field and the relevant application will automatically open (the camera, the audio recorder or the video camera).—Once finished—Press on ‘OK’ and the file just captured will be attached to the form’s File Upload field.—Submit the form—the user can check the user’s email inbox and the user will receive the submission, which includes the captured file of one embodiment.

#### Box Push Integration Form Builder Pdf Mapping:

[0177] FIG. 22 shows for illustrative purposes only an example of Salesforce push integration form builder pdf mapping of one embodiment. FIG. 22 shows Salesforce push integration form builder pdf mapping 2200 to push data 2210 to an integrated platform. Step 1—choose platform 2220 allows a user to choose/remove a platform to push data to: 2230. For example the user may choose an application platform by clicking on the input box to place an “X” 2235 for the choice selection of one embodiment.

[0178] FIG. 22 shows the application platform push integration form builder pdf mapping. The application platform offers secure content management and collaboration for individuals, teams and businesses, allowing secure file sharing and access to the user’s files online. By using the customized customer relationship management platform method and devices the Push option can automatically update the application platform account.

[0179] Following are steps to integrate with the application platform. 1—Open the form to integrate in the Form Builder. 2—Go to “Properties Panel”>“Form” tab>“Settings”>“Integrations” Category>“Push notification”—Press on “Set Notifications”  
3—in the first wizard step: Choose Platform, select the application platform. 4—in order to push data to the application platform first “Authenticate with the application platform”. 5—Once authenticated enter the application platform Parameters:

File Prefix—This field is not mandatory, but can chose to add a prefix to the file name created in the application platform. Folder—Enter a folder name, which will be set up in the application platform account and will store the entries. This field is mandatory. 6—Turn on the notification toggle and Press on “Finish”. Example for how the integration works, now that the form is configured to integrate with the application platform—a user will fill in the form and then: 1—Once the user enters the user’s application platform account find the folder configured in the push process and the entries will be stored under it.

#### Push Settings Tab:

[0180] FIG. 23 shows for illustrative purposes only an example of push settings tab of one embodiment. FIG. 23 shows a push settings tab 2300 on a push settings tab

webpage 2310. The user makes selections for an authentication list & status 2320 of a platform for example Salesforce . . . Salesforce @ FormTitan 2325 to receive automatically Push/Get Logs 2330 from the integrated Salesforce platform.

[0181] Other push features include error messages for skip in push, Map fields in get—made wide like in push, No override of empty fields in push, Redirect only after push is complete, All option added to Push Update, error message in Push logs: FT Request Limit, filter in Push Logs, Form Name in Push Logs, Ignore mandatory and validations in Custom Push, Push logs available in my submissions, Push mapping Condition, Redirect in Custom Push and Get, integration—Push to VoIP services, and Values of Hidden fields now shown after custom Push of one embodiment.

[0182] FIG. 23 shows Push Settings Tab wherein “My Account” gives a user quick access to settings and tools for managing the user’s customized customer relationship management platform method and devices account. It is divided into nine tabs and “Push Settings” tab is the eighth. This tab is divided into 2 main areas: 1—Push Authentication List & Status—on the left and 2—Push Logs—on the right.

#### Push Authentication List and Status:

[0183] FIG. 24 shows for illustrative purposes only an example of push authentication list and status of one embodiment. FIG. 24 shows a push authentication list and status 2400 with which a user can check the push authentication status 2410 of various platforms they may have chosen. The platforms can include for example Salesforce . . . Salesforce @ FormTitan 2325, and Salesforce Sandbox 2424. The push authentication status 2410 provides a user with a selection to email me errors 2415. The push authentication list and status 2400 provides information and selections for a user showing the status of platforms including not authenticated 2445 or authenticated 2440. The user can select features to revoke access 2450 or grant access 2460 to a platform of one embodiment.

[0184] FIG. 24 shows a Push Authentication List & Status including a list of all the 3rd party integrations FormTitan provides, also called “Push integrations” (because a user can “push” the data out). Push integrations can include for example Salesforce and other application platforms. In the list a user can see the 3rd parties currently authenticated. A user using the list features can authenticate or revoke access from an application platform. Additionally is shown an “Email me errors” checkbox, which is turned on by default. This makes sure that if errors occur while users submit a form that is integrated the user will receive an email containing this error of one embodiment.

#### Push Logs:

[0185] FIG. 25 shows for illustrative purposes only an example of push logs of one embodiment. FIG. 25 shows push logs 2500 a user can select to check data pushed. The push logs 2500 can provide a user with a listing of data pushed to for example Salesforce 2510 of one embodiment.

[0186] FIG. 25 shows Push Logs. The table is called a ‘Push/Get log’. It contains all the integration action rows for both a user’s push and get integrations. Each time an action is taken—either get or push—an action row is created in the log containing: Form Id: what form the action belongs to, Date time: Date of the action, IP: the form filler IP, Sub-

mission ID: the submission Id the action belongs to. This is also a link that displays the submission in “My submissions” page. Other integration action rows include Method: Integration party, for example: Salesforce, Status code: Success or error, Message: Each message contains: the object used in the action (for example ‘Account’ or ‘Contact’), the action that was taken (Create/Update/Delete . . . ), and the error description including a link to the object in Salesforce/or another 3rd party, Resubmit: When an action fails and there is an error—then data is not synced with the 3rd party like it should. The information is not lost and after a user makes the required fix in the form—the user can press on this “Resubmit” link and have the missing data added.

#### Personalized Preventive Care and Wellness Plan:

[0187] FIG. 26 shows for illustrative purposes only an example of personalized preventive care and wellness plan of one embodiment. FIG. 26 shows a personalized preventive care and wellness plan 2600. A user can for example open a personalized preventive care and wellness plan description 2610 to review and revise as they see fit of one embodiment.

[0188] The customized customer relationship management platform can be used in health care user applications. The health care industry uses a great deal of forms from initial patient intake, exams, lab test, diagnostics and so forth. The customized customer relationship management platform form builder can reduce workloads by automating data collection using form builder responsive forms that gather the data directly rather than have manual entry of the data. Many health care companies have multiple locations and even nationally distributed locations. Some of that same location may use a different 3rd party application which further complicates data collection. In addition the federal requirements for maintaining electronic health records need standardization within a company to prevent non-compliance issues. The customized customer relationship management platform functionality and integration capability can allow a company to standardize data collection while maintaining different 3<sup>rd</sup> party application thereby saving the replace cost of the applications.

#### HIPAA, GDPR and Other Compliances are Supported:

[0189] FIG. 27 shows for illustrative purposes only an example of HIPAA, GDPR and other compliances are supported of one embodiment. FIG. 27 shows HIPAA, GDPR and other compliances are supported 2700 in the customized customer relationship management platform method and devices. A preview 2710 of potential health issues is provided to allow a user to comply with regulations and suggested health care programs described in the Health Insurance Portability and Accountability Act (HIPAA) and General Data Protection Regulation (GDPR) to keep a patient information private. The preview 2710 includes for example a series of questions the user can answer for example how much over the past month felt tired: yes, no, sometimes 2720 very mild, severe 2730 and are you limited in any way due to a physical, emotional, or mental problem: yes, no, sometimes 2740 of one embodiment.

[0190] HIPAA, GDPR and other compliances support allows health care companies to comply with the myriad of compliance regulations using a single source product for form creation or conversion, not have to change existing

hardware and software and standardize their operations across multiple facilities regardless of location using the customized customer relationship management platform functionality and integration capability.

#### PDF Mapping Settings:

[0191] FIGS. 28-32 show for illustrative purposes only an example of pdf mapping settings of one embodiment. FIG. 28 shows for illustrative purposes only an example of pdf mapping settings of one embodiment. FIG. 28 shows pdf mapping settings 2800 on a mapping field 2810 webpage. The mapping fields 2810 webpage includes a selection to submit options 2820 for settings 2840 including for example a general 2845 selection to preview before submit 2830 of one embodiment.

#### PDF Mapping Settings General:

[0192] FIG. 29 shows for illustrative purposes only an example of pdf mapping settings general of one embodiment. FIG. 29 shows pdf mapping settings general 2900 including submit options 2820 of the general 2845 features including preview before submit 2830, print and save 2931, interactive mode 2932, print page number 2933, format 2934 for example A4, header 2935 with an instruction to configure . . . is not set showing it has not been configured, footer 2936 with an instruction to configure . . . is not set showing it has not been configured, and auto preview 2937 of one embodiment.

#### PDF Mapping Settings Buttons and Labels:

[0193] FIG. 30 shows for illustrative purposes only an example of pdf mapping settings buttons and labels of one embodiment. FIG. 30 shows a pdf mapping settings buttons and labels 3000 feature including the submit options 2820 and a selection menu for buttons & labels 3020. The buttons & labels 3020 selection menu includes finish button 3030, close button 3040, title 3050, signature 1660 with an instruction to click here to sign, and file upload 3070 with an instruction to click here to upload a file of one embodiment.

[0194] Additional settings buttons include features including Add and remove buttons in repeated section, Add script after button completes its task, Allow/Disable condition result options for Checkbox and Radio button, Condition button type allows to manipulate section tabs, Confirmation Mode on button, Disallow items of Picklists in Dropdown, Radio button, Checkbox/Multiple dropdown, Draft button—Hide confirmation, Excess buttons removed from repeated section, Move to a specific page using the Salesforce Action button, button type: Redirect, Page break with flexible next/prev buttons, Repeated section—Show remove button for last, Styles for buttons in Table element including alignment, Text direction buttons in HTML editors, Button—Reset Signature, Option to Ignore Empty Conditions in Action Button—Get, Go to page Button based on Hidden, Button Type: Condition, On click event added to Button, Add and remove buttons in repeated section, Button—Reset Signature, Button Type: Condition, button type: Redirect, On click event added to Button, Repeated section—Show remove button for last, Condition button type allows to manipulate section tabs, Confirmation Mode on button, and Text direction buttons in HTML editors of one embodiment.

#### PDF Mapping Settings Email:

[0195] FIG. 31 shows for illustrative purposes only an example of pdf mapping settings email of one embodiment. FIG. 31 shows a pdf mapping settings email **3100** feature that includes the submit options **2820** selection. The pdf mapping settings feature email **1070** includes a custom file name **3130**, send to owner **3140**, send to additional **3150**, and send to form emails **3160** of one embodiment.

#### PDF Mapping Settings Condition:

[0196] FIG. 32 shows for illustrative purposes only an example of pdf mapping settings condition of one embodiment. FIG. 32 shows a pdf mapping settings condition **3200** feature including the submit options **2820** selection. A condition **3220** feature includes menu selections to generate condition **3230**, process upon payment only **3240** and a page condition **3250** of one embodiment.

[0197] PDF Mapping: settings. Configuring PDF Settings. No matter what type of PDF a user adds to a form (Basic, Dynamic, or Auto PDF)—the user will need to configure its settings. While the Basic and Dynamic PDF settings are the same, the Auto PDF has a slightly different set of properties. The Basic/Dynamic PDF Settings 1—Press on the “settings” icon to configure the PDF settings. A Modal window will open then containing the settings, organized in 4 categories:—General—Buttons and Labels—Email—Condition. Please notice that the first category will be open by default.

[0198] General properties:—Preview before submit: turning on this checkbox will allow a user to present the form filler with a preview to a soon-to-be-generated PDF. It will display the template the user created and the data the form filler entered in it. So, in fact, when the form filler hits the submit button a pop up window will open, containing this preview. Only after the form filler presses on the “Finish” button in the preview window it will actually submit the form. The PDF preview window contains 4 icons in the top right corner to help the form filler: Plus (for zooming in), minus (for zooming out), clear (for clearing the zoom) and Print (for printing the PDF).

[0199] Print and save: turning on this checkbox will add a saving functionality to the print icon. In this case, pressing on the Print icon will open a dialog for print, and once the printing is approved—it will also submit the form. —Interactive mode: This option will only be available when the ‘preview before submit’ checkbox is turned on. The interactive mode will allow form fillers to add or edit data inside the PDF preview window. Once this checkbox is turned on a user will be able to choose the fields to make the fields selected to be interactive. Print page number: If the PDF has more than one page a user can turn on this option to add a numbers at the bottom of each page. Format: The default PDF format is A4, however a user can choose to change it to Letter. Header: This option can allow a user to add a header to the PDF using the editor and HTML. Footer: This option can allow a user to add a footer to the PDF using the editor and HTML. Auto preview: This option will automatically open the preview window after the form loads, so the form filler will be able to “skip” filling in the form.

[0200] Buttons & Labels properties:—Finish/Close button text: if a user decides to turn on the ‘preview before submit’ option, the user can change the text on the buttons at the bottom of it. Title: If a user decides to turn on the ‘preview before submit’ option, the user can add a title text to this

window. Signature: When a Signature field is added to the PDF and made interactive (by turning on the ‘interactive mode’ and adding the signature field to the list of interactive fields), it will have a label below it, containing the default text: “Click here to sign”. A user can enter a different text in this input box to replace it. File upload: When a File upload field is added to the PDF and made interactive (by turning on the ‘interactive mode’ and adding the File Upload field to the list of interactive fields), it will have a label below it, containing the default text: “Click here to upload a file”. A user can enter a different text in this input box to replace it.

[0201] Email properties: Custom file name—a user can enter a name for the PDF being now created. This will be the file name of the PDF attached to the email. Send to owner/Send to additional/Send to form emails: Checkboxes that determine where this paper form will be sent to. Condition Properties: Generate Condition: if a user decides to turn on this Checkbox the user will be allowed to add a condition by which this PDF will be generated. Only if the condition terms are met then the PDF will be generated. Process upon payment only: turning on this checkbox will make sure that the PDF will be generated only if the payment is executed. Page Condition: By turning on this option a user can set a condition that will make PDF pages hide/show based on the form filler’s input. PDF Mapping Form Settings. This Category holds the configuration of the PDF Mapping feature. 1—Map Fields by pressing on the “Map Fields” button” a user can map the online form on to a paper form scan. Once the window opens a user will need to place the online form fields above a paper form, in the right places.

#### Form Builder Optimization Features:

[0202] FIG. 33 shows for illustrative purposes only an example of form builder optimization features of one embodiment. FIG. 33 shows form builder optimization features **3300** includes CRO **3310**, Push Integrations **3320**, Heatmap **3330**, Chat **3340**, Payment Integrations **3350**, Sentiment Analysis **3360**, A/B Testing **3370**, Form Conditioning **3380**, and Auto Translate **3390** of one embodiment.

[0203] Optimization Form Settings. This Category holds the properties related to the optimization of the form. 1—Chats. This property, when turned on, allows a user to activate a chat in a form and talk to form fillers right then and there. Once a user turns on this option the chat becomes allowed, a user will need to start the chat from the “My Forms” page.

[0204] Chat 2—Heatmap: This property allows a user to create a heatmap based on the form. Once a user turns on this checkbox, a user can start recording the visitors’ movements on a form. This creates a graphic image made of colors a user can then analyze. In order to view the heatmap created a user will need to click on the “View heat map” link in the “My Forms” page. Heatmap 3—Randomize this property allows a user to show the form elements in random order each time the form is loaded by the user. 4—Auto Translate This feature adds a translation option to a published form. Once a user, with a language different from the one set in the form and opens it the user will have an option to translate it to a selected language or use it as is.

[0205] 5—Prediction: The following property allows a user to use the sentiment analysis feature. With this feature a user can try to understand the thoughts and intentions of form fillers. 6—Type of form: The following property lets a user categorize a form, according to the form types specified

in the list. The form type is important because it has a direct influence on the CRO tips a user will get. 9—Autocorrect for Mobile. This option turns on the Mobiles “auto correct” for mobile devices. This option is turned off by default since auto correct in mobiles can interfere with the form filling process. 10—Show Valid Indicator: This option is turned on by default. It makes sure that each time a form filler enters a form field correctly; a green V icon will appear next to it, to indicate approval. A user can hide these indicators by turning off this checkbox. 11—Don’t save the data: Data entered in a form is stored by default, which later allows a user to export it, analyze it etc. However, if a user wants to stop saving the data the user can turn off this option. 12—Generate barcode: This property allows a user to generate a barcode.

#### Custom Translation:

[0206] A user can create an online form that needs to be displayed in several languages using an Auto Translate optimization that will enable the online form to be presented to foreign users with in their native language. However, in some cases a user may find the automatic translation insufficient or not precise enough. A user may be using technical terms, or even simple words with multiple meanings and need to translate things manually. A custom translation feature will allow the user to use this feature to control all the labels, captions, user tips, button text in a form, and can create different translations to as many languages as the user selects.

[0207] Following is an example of how to add custom translation to a simple form in French: 1—Enter the form builder and create a new form. 2—Drag a textbox and change its label text to: “Nom de la compagnie” which means “Company Name”. Change the text on a button to “soumettre le formulaire”, which means “submit form”. 3—in order to have the form translated a user must first declare the form’s original language, in this case: French. Go to “Properties” panel>“Form” tab>“Settings” option>“Personalization” category. 4—Choose “French” in the Language drop down. 5—The user can add the custom translation. Go to: “Properties” panel>“Form” tab>“Settings” option>“Optimization” category.

[0208] 6—Turn on the “Custom Translation” checkbox and then press on the “Edit” button below it. 7—Choose a language to translate the form to. For example: English. 8—The fields of the form will now be displayed in small tables, each containing the current field text (in French), and a user replace these texts with the English version. For example: a user can change “soumettre le formulaire” text on the button to “Submit this form now” which has a slight difference in meaning. 9—Press on “Apply” and save the form. 10—Now the user can publish the form as URL and see the result: Whenever a translation feature is used in the form (Auto or Custom) a pale blue strip will appear at the top of the web page, saying that the page has been translated. It also gives the user the opportunity to view the form in its original language, if the user chooses to of one embodiment.

#### Form Builder Features Characteristics:

[0209] FIG. 34 shows for illustrative purposes only an example of form builder features characteristics of one embodiment. FIG. 34 shows the customized customer relationship management platform method and devices objec-

tives including for example Create beautiful forms 3400, Do more with your data 3410, Add brains to your forms 3420, Reduce form abandonment by chatting with your users 3430, Protect your forms 3440, Map your paper form to an online form 3450, Target your audience 3460, Optimize like a pro marketer 3470, Gain more insight 3480, Make money with your forms 3490, and Add workflow to your documents 3495 of one embodiment.

[0210] Each of the form builder features and elements serves a functional purpose. FIG. 34 shows a brief description the characteristics of the features and elements and the general functional purpose behind those features and elements.

#### Other User Applications:

[0211] FIG. 35 shows a block diagram of an overview of other user applications of one embodiment. FIG. 35 shows other user applications 3500 including health care 3510, onsite inspections 3512, and human resources 3514. Additional applications in an enterprise environment 3516 include judicial branches—case loads, conviction results, witness interviewing 3520, factory—employee production levels, sales, operations costs & suppliers 3525, transportation—bus, trolley, air cargo, trucking, shipping, rail passenger and cargo 3530, military—logistics, personnel, veterans services 3540, brick and mortar retail chains—product sales, suppliers 3550, law enforcement—officer response times, ticketing frequency & review, arrest rates 3560, colleges and universities—class loads, student performance for professors, operations costs & revenues vs. enrollment 3570, postal services and private package delivery—logistics, product and services sales 3580, and federal, state and local government services, logistics, personnel performance, public services 3590 of one embodiment.

[0212] FIG. 35 shows other user applications where the customized customer relationship management platform functionality and integration capability can be an added asset to reduce cost and improvement management tools. The other user applications can include Health Care, Onsite Inspections, and Human Resources. Many industries in the Enterprise Environment can adapt forms or create forms to improve data collection including Judicial Branches—Case Loads, Conviction Results, Witness Interviewing; Factory—Employee Production Levels, Sales, Operations Costs & Suppliers; Transportation—Bus, Trolley, Air Cargo, Trucking, Shipping, Rail Passenger and Cargo; Military—Logistics, Personnel, Veterans Services; Brick and Mortar Retail Chains—Product Sales, Suppliers; Law Enforcement—Officer Response Times, Ticketing Frequency & Review, Arrest Rates; Colleges and Universities—Class Loads, Student Performance for Professors, Operations Costs & Revenues vs. Enrollment; Postal Services and Private Package Delivery—Logistics, Product and Services Sales; and Federal, State and Local Government Services, Logistics, Personnel Performance, Public Services.

#### Other User Applications Enterprise Environment:

[0213] FIG. 36 shows a block diagram of an overview of other user applications enterprise environment of one embodiment. FIG. 36 shows other user applications 3500 in an enterprise environment 3516 include online sales and distribution logistics 3620, cable networks—cable services, programming monitoring, on-air advertising and marketing

**3630**, broadcast networks—programming monitoring, on-air advertising and marketing **3640**, and communications providers—internet, voice, video and streaming, advertising and marketing **3650** of one embodiment.

[0214] FIG. 36 shows additional other user applications in an Enterprise Environment including Online Sales and Distribution Logistics; Cable Networks—Cable Services, Programming Monitoring, On-Air Advertising and Marketing; Broadcast Networks—Programming Monitoring, On-Air Advertising and Marketing; Communications Providers—Internet, Voice, Video and Streaming, Advertising and Marketing; Food Services, Restaurant Chains—Menu Item Ordering Monitoring, Ingredient Purchasing & Suppliers; and News Services—Reporters, Camera Crews, Marketing and On-Air Advertising.

#### Bi-Directional Voice Command and Interactive Form Building Elements:

[0215] FIG. 37A shows for illustrative purposes only an example of bi-directional voice command and interactive form building elements of one embodiment. FIG. 37A shows bi-directional voice command and interactive form building **3700**. The customized customer relationship management platform method and devices is accessible through a user smart phone **3710** internet connection to the customized customer relationship management platform method and devices digital servers using a FormTitan digital application installed on a user digital device including the user smart phone **3710**. In this example the user opens form builder elements **3720** while creating interactive user's existing pdf form mapping fields **3724**. The FormTitan digital application includes an audio text reader to transmit an audible question “which form builder element?” **3722** to the user **3730** through a hands-free ear piece **3732**. In response to the audible question regarding the contact us **3726** mapping fields selection the user **3730** conveys the user's verbal response is “text box” **3734**. The user's verbal response is received by the smart phone **3736** FormTitan digital application. The FormTitan application enters a textbox onto the user's existing pdf form mapping field display **3738**. The FormTitan application selects a textbox from the form builder elements per the user's verbal command **3740** of one embodiment.

[0216] FIG. 37A shows a customized customer relationship management platform user employing the voice element of the form builder. The mini mode has converted the form builder customization screen to fit on the user smart phone screen. The voice has audibly read the elements available and asks the user “which form builder element?” The user listens to the possible selections using for example a Bluetooth device. The user then responds speaking into the smart phone microphone with the “textbox” selection. Upon receiving an element selection the form builder places a textbox element in the body of the form of one embodiment.

#### Bi-Directional Voice Command and Interactive Form Building Data:

[0217] FIG. 37B shows for illustrative purposes only an example of bi-directional voice command and interactive form building data of one embodiment. FIG. 37B shows bi-directional voice command and interactive form filling **3705** using the user smart phone **3710** to fill in the user's existing pdf form mapping fields **3724** contact us **3726** data.

The user **3730** hears through the hands-free ear piece **3732** a FormTitan application process to the next data entry for example first name **3750**. The FormTitan digital application an audio text reader transmits an audible question “first name?” **3752**. A user's verbal response is “Bob” **3762**. The user's verbal response is received by the smart phone **3736** and the FormTitan application enters “bob” in the textbox **3766**. Additionally the FormTitan application shows “Bob” in the user's existing pdf form mapping fields display **3768**. The FormTitan application automatically displays the first name entry **3770**. The contact form shows the first name “Bob” on the contact us form **3772** in the first name text box of one embodiment.

[0218] FIG. 37B shows in another embodiment the voice element could be asking a user form filler for his name “first name” according to the data element on the form. The user form filler responds with his first name “Bob”. The validated audible response is converted into text and entered in the designated textbox on the responsive form. In another embodiment the user form filler could be speaking another language and the Auto Translate feature can translate the response into the language designated on the form of one embodiment.

#### Bi-Directional Voice Command and Interactive Information from Forms:

[0219] FIG. 38 shows for illustrative purposes only an example of bi-directional voice command and interactive information from forms of one embodiment. FIG. 38 shows bi-directional voice command and interactive information from forms **3800**. The user makes a verbal query request **3870**, “what events are coming up?” **3850** on the user smart phone **3710**. The FormTitan application displays the user's calendar of events **3830** on the smart phone. The FormTitan application searches the user's calendar of events **3830** for the next event data **3832**. The FormTitan application displays the user's calendar of events **3830** on the smart phone **3710** “august 22 board meeting” **3825**. The FormTitan application using the text reader transmits an audible message to the user of the next event “august 22 board meeting” **3860**. The user **3870** using a hands-free ear piece **3880** hears the FormTitan application audible message of one embodiment.

[0220] FIG. 38 shows a customized customer relationship management platform user employing the voice element of the form builder to query a calendar of events. The user speaks the question “what events are coming up?” into the microphone of his smart phone which is connected to the platform. The voice element reads aloud the next event on the calendar of events resulting from the voice activated query. The voice element responds audibly that the next event is an “August 22 Board Meeting”.

#### Integration Example with Salesforce:

[0221] FIGS. 39-49 show for illustrative purposes only an example of integration example with Salesforce of one embodiment.

#### Integration Example with Salesforce:

[0222] FIG. 39 shows a block diagram of an overview of integration example with Salesforce of one embodiment. FIG. 39 shows an integration example with Salesforce **3900**. The Salesforce app **3910** is accessed using the integrated Salesforce platform. The following are step by step instructions to populate contacts of a chosen account **3920**. Step 1—simply enter Salesforce, add a few contacts under several accounts and be sure to fill in first name and last name for

each one **3930**. Step 2—enter the form builder and open the form created in example #1 and drag a section element **3940**. Step 3—drag 2 textboxes inside the section, and place them one above the other **3950** of one embodiment.

#### Integration Salesforce Object Settings:

**[0223]** FIG. 40 shows for illustrative purposes only an example of integration Salesforce object settings of one embodiment. FIG. 40 shows a process to map Salesforce fields **4000**. A user makes a connection (default) Salesforce1@FormTitan.com **4010**. The user opens Salesforce object settings (account) **4020** and proceeds to select an object **4030** in an account **4040** and enter a comment **4050** of one embodiment.

#### Integration Setting Salesforce Section:

**[0224]** FIG. 41 shows for illustrative purposes only an example of integration setting Salesforce section of one embodiment. FIG. 41 shows the selection of section **830** that displays a choose account name select drop down **4110** for a user to select a name from a section selection **4120** including a contact last name **4130** of one embodiment.

#### Integration Setting Salesforce Fields:

**[0225]** FIG. 42 shows for illustrative purposes only an example of integration setting Salesforce fields of one embodiment. FIG. 42 shows a webpage where a user can select a choose account name select drop down **4110** to select an account name. Under the form **4210** a user proceeds to settings [?] **4220** to set notification **4230**. Once the notification is enter the mapping is set **4240** in the map fields **4250** of one embodiment.

#### Integration Get from Salesforce:

**[0226]** FIG. 43A shows for illustrative purposes only an example of integration get from Salesforce of one embodiment. FIG. 43A shows a Get from Salesforce **4300** feature for integrated use with the Salesforce app **3910**. When the Salesforce app **3910** is selected a process on the customer relationship management platform network **210** of FIG. 2 performs an action to authenticate with Salesforce **4320** a production **4321** operation. The customer relationship management platform network **210** of FIG. 2 finds Salesforce1@FormTitan.com authenticated **4322**. The user then selects for example 1 account **4330** or 2 account **4331** of one embodiment.

#### Integration Map Salesforce Fields:

**[0227]** FIG. 43B shows for illustrative purposes only an example of an integration map of Salesforce fields of one embodiment. FIG. 43B shows map Salesforce fields **4000** including Salesforce object settings (contact) **4350**. The select an object **4030** includes the contact **1230** and a comment **4050** entry textbox of one embodiment.

#### Integration Map Salesforce Object Settings:

**[0228]** FIG. 44 shows for illustrative purposes only an example of an integration Salesforce map object settings of one embodiment. FIG. 44 shows a webpage for map Salesforce fields **4000** including Salesforce object settings (contact) **4350**. The user can select an object **4030** for example the contact **1230** where the user can enter a comment **4050**. The user can see displayed that a condition is set **4450** when

the user set conditions first **4460** referring to a first name and the display indicates conditions is set **4470** and the user can set a limit **4480** as a condition of one embodiment.

#### Integration Salesforce Condition:

**[0229]** FIG. 45 shows for illustrative purposes only an example of integration Salesforce condition of one embodiment. FIG. 45 shows a Salesforce condition **4500** webpage with an account ID drop down **4510** to select an account that meets a condition in a drop down for example an equals drop down **4520** in a #2 account drop down **4530** and matching account ID drop down **4510** selection of one embodiment.

#### Integration Salesforce Mapping Corresponding Fields:

**[0230]** FIG. 46 shows for illustrative purposes only an example of integration Salesforce mapping corresponding fields of one embodiment. FIG. 46 shows Salesforce mapping **4600** including a filter: first **4610** and how it is to show: please choose drop down **4660** using the drop down. A form field **4620** first name **4650** selection from a Salesforce field **4640** contact first name **4630** can be selected as the filter condition of one embodiment.

#### Integration Salesforce Mapping Contact:

**[0231]** FIG. 47 shows for illustrative purposes only an example of integration Salesforce mapping contact of one embodiment. FIG. 47 shows a Salesforce mapping—#2.1 to select a contact **4700** filter: last **4705**. A form field **4620** contact last name **4130** can be selected from a Salesforce field **4640** last name **4740** of one embodiment.

#### Integration Salesforce Get Contact:

**[0232]** FIG. 48 shows for illustrative purposes only an example of integration Salesforce get contact of one embodiment. FIG. 48 shows a Get from Salesforce **4300** operation using the Salesforce app **3910**. The authenticate with Salesforce **4320** can for example use a production drop down **4830** to obtain a Salesforce1@FormTitan.com authenticated **4322** notice. The user selects 1 account **4330** or 2 account **4331**. In this example the user selects the 2 account **4331** drop down to get a 2.1 contact **4870** to receive a notification: **4880** of one embodiment.

#### Integration Salesforce Email:

**[0233]** FIG. 49 shows for illustrative purposes only an example of integration Salesforce email of one embodiment. FIG. 49 shows a user selecting from the elements **550** multiple input **890** feature email **1070**. An email input box **4930** is used by the user to enter an email address. A good indicates the quality of the email entry of one embodiment.

**[0234]** An integration example with Salesforce wherein a user uses a Salesforce app. Following are step by step instructions to populate contacts of a chosen account: 1—In order for this example to work a user will need to make sure there is at least one sample data in the Salesforce “account” object. Simply enter Salesforce, add a few contacts under several accounts and be sure to fill in first name and last name for each one. 2—Enter the form builder and open the form created in example #1. Drag a section element. 3 drag 2 textboxes inside the section, and place them one above the other.

[0235] Integration Salesforce Object Settings:

[0236] Salesforce Push Category—Salesforce Object Settings When the “Map Salesforce fields” window opens for the first time, only two configuration categories will be visible: the “Connection” category and “Salesforce Object Settings” category. Even though this is the second category it will be the one that is open by default, since it is the basis for the entire integration—this is where a user can choose which Salesforce object will be dealt with in this integration line. So go ahead and choose an object. This category contains three fields: 1—‘Select an object’ dropdown—this dropdown contains all of the Salesforce objects—just select the one to sync with. This is a mandatory field—the process cannot proceed without choosing an object. 2—‘Comment’ textbox—This field is not required and will not be visible to the form filler—it is for the user purposes only. The comment allows a user to add a short description to the integration line being created and help the user recognize it in the future. 3—“Use in custom button” checkbox—Turning on this checkbox will allow a user to use a Salesforce Action button as trigger instead of the regular submit button of one embodiment.

[0237] Integration Setting Salesforce Section:

[0238] A Salesforce action button can easily be added to a form to get/push data right then and there, without having to submit the form. However, when setting the button to act on a repeated section it applies to all of the data in all of the repeated lines. If a user wants to have a form filler press on the custom button in a specific repeated line and have it apply to this line only the user can follow these steps: 1—Create a new blank form. 2—Drag a section element and make it wider. 3—Drag a textbox, a numeric and a button into the section. 4—Change the textbox label to: account Go to ‘Properties’ panel>‘Element’ tab>‘Settings’ option>‘Basic’ category>Label. 5—Change the numeric label to num. 6—Change the button text to: push this item now. 7—Select the section and make it Repeated. Go to ‘Properties’ panel>‘Element’ tab>‘Settings’ option>‘Basic’ category>Repeated.

[0239] 8—Now configure the Salesforce push integration ‘Properties’ panel>‘Form’ tab>‘Settings’ option>‘Salesforce integration’ category>‘Push to Salesforce’—Press on the ‘Set notification’ button—Authenticate with Salesforce—Choose the account object: Account—Turn on the ‘Use in custom button’ checkbox—in the ‘Action’ category—leave the ‘Create’ option selected—Map the fields: Account Name-->account Num>num—Press ‘apply’, then ‘Finish’ and save the form. 9—Select the ‘button’ and change its type to ‘Salesforce Action’. 10—Press on the ‘Set’ button to have the integration line work with your custom button. 11—Open the ‘process push or get’ dropdown and select the Salesforce action. 12—Press on ‘Add’ and it will be added to a table above. 13—Turn on the ‘Execute per repeated item’ checkbox. 14—Press ‘Apply’ and save your form.

[0240] 15—Now it’s time to test the form:—First select the Section and change the visible items to 2 and turn on the ‘Populate items’ checkbox—Publish it as URL—Enter the first item line in the section: an account name and number: test 1, 1 and test 2, 2.—Press on “Add” and enter another item line with data: test 3, 3—Choose a specific line (for example second line) and press on the ‘push’ button in that line.—Check the Salesforce account and the user will see that only that line of data was added as a new record.

[0241] Integration Setting Salesforce Fields:

[0242] Control which fields on the form will be saved in Salesforce based on action FormTitan Push integration with Salesforce allows a user to create new records in Salesforce objects as well as Update existing records. For example:—To create a lead in Salesforce—and default the company field if its empty and a user can prevent an update of the company name if the action is updated.

[0243] A bi-directional Salesforce form. FormTitan allows a user to:—Read data from Salesforce and have the form fields populated straight from the Salesforce objects in real-time (Aka Get).—Write data to Salesforce objects in order to create a new record, update an existing, or even delete a record (Aka Push). A user can combine the Get and Push to Get data from multiple Salesforce objects and Push the data to multiple Salesforce objects and do it all in a single form.

[0244] Integration Get from Salesforce:

[0245] Salesforce Get Category—Salesforce Object Settings The first time this window will open, only the first two categories will be visible: “Connection” and “Salesforce object setting”. The latter will be open by default, since it is the starting point of the integration—this is where a user can choose a Salesforce object. So even though this is the second category, it will be the one that is open by default. This category contains three fields: 1—‘Select an object’ dropdown—this dropdown contains all of the Salesforce objects—just select the one object to sync with. This is a mandatory field—the process cannot proceed without choosing an object. 2—‘Comment’ textbox—This field is not required and will not be visible to the form filler—it is for the user’s purposes only. The comment allows a user to add a short description to the integration line being created and help the user recognize it in the future. 3—“Execute on form load” checkbox—Turning on this checkbox will execute the get action and populate the fields when the form is loaded. This is of course not a mandatory field and it is turned off by default. When it is not turned on the user will be asked to set a condition (in the next category) by which the data will be drawn from Salesforce of one embodiment.

[0246] Integration Map Salesforce Fields:

[0247] Forms Integration with Salesforce allows a user to read data from Salesforce and update it. A user can populate the account name and account number from Salesforce using Get integration. The user can configure a push integration so the form filler can update these details and have them sync in real-time. Fields Order in Salesforce Form can be used for creating a new form that for example is called ‘Salesforce’ and it helps the user create a form based on a Salesforce object in an automated way. When the user chooses this feature the user needs to authenticate with Salesforce, select an object and pick the fields for the form. Once completed the form is generated for the user on the form canvas and the user can adjust the styling. A user can to define the fields order prior to creating the form.

[0248] Integration Salesforce Mapping Corresponding Fields:

[0249] Mapping Get with Salesforce lookup fields includes a feature to fetch a Lookup name from Salesforce. A user can only map to its record ID, and since the ID is of no interest to form fillers a user can create formula fields in Salesforce to get the Lookup’s name, or use more ‘Get’ actions to reach this data. A user can map the same Lookup and reach any related field three levels deep. (A field in the

same level, and 2 levels above it). The action can be done in the same Get action, without adding formula fields, without additional Get actions, saving the user time and API calls.

[0250] For example: The ‘Case’ object in Salesforce contains a ‘Contact name’ lookup, that shows which contact is the owner of this case. The Case object Contact name lookup avoids a user when using the “Get” integration mapping of only fetching the Contact ID, having to create a formula that was pulling the account name based on this account ID. A user can use the same mapping object in the Salesforce integration to choose any of the related fields that the lookup field possess and have it nested up to 3 levels deep! For example: when setting up Get integration mapping to ‘Contact’ object fields, a user can also fetch the Parent Account name (which is a Lookup) from within the ‘Contact’ object: allowing a user to go from Contact to accountid to parentaccountid and then to account name of one embodiment.

[0251] Integration Salesforce Mapping Contact:

[0252] Create multiple contacts and cases in Salesforce. Utilize the section control to create multiple Salesforce objects. A user can allow form fillers to create an account through the form to allow the form filler to create contacts under this account and to allow a form filler to create cases under a contact. When building the form use the Section element to create a multiple items container to make the section repeated, so the form filler can add items. A user can configure the Salesforce Push integration so for example 3 objects are associated in a parent-child relationship then test this form, once it is set up of one embodiment.

[0253] Integration Salesforce Email:

[0254] a PDF based on Salesforce data and send it via custom email. PDF: PDF mapping allows a user to take the values entered in the form fields and generate a PDF document that contains them. A user can use the PDF mapping PDF based on Salesforce data wherein PDF draws its field values from the form. It makes no difference if the form filler entered the field values or if the field values were populated from Salesforce automatically—the value found in the field is the one used. A user can create a PDF based on data from Salesforce by setting up a Get integration in order to populate the fields in the form.

[0255] Send PDF via custom email allows a user to configure the PDF mapping using three checkboxes that allow the user to have it sent via mail to the form owner, form filler and additional emails. If the ‘form filler’ checkbox is turned on, for example, then the PDF document is added as an attachment to the email currently set up for the form filler. Please note that there is no email sent to the form filler by default, so a user will need to make this option active. In addition, it does not matter if the user uses the default email or the customized email because the PDF will be attached to both.

#### PDF Block Condition Elements:

[0256] FIGS. 50-60 show for illustrative purposes only an example of pdf block condition elements of one embodiment.

#### PDF Block Condition Elements:

[0257] FIG. 50 shows for illustrative purposes only an example of pdf block condition elements of one embodiment. FIG. 50 shows a label phone 5000 element selection where a user enters a phone number in a phone input text box

5010. The phone input is converted into a label automatically 5020 of one embodiment.

#### PDF Block Condition Dynamic Mode:

[0258] FIG. 51 shows for illustrative purposes only an example of pdf block condition dynamic mode of one embodiment. FIG. 51 shows mapping fields 2810 where dynamic mode is shown on 5110. A notice shows mapping is set map fields 5120 wherein the mapping is set confirms that the dynamic mode is on 5130 of one embodiment.

#### PDF Block Condition Pdf Settings:

[0259] FIG. 52 shows for illustrative purposes only an example of pdf block condition pdf settings of one embodiment. FIG. 52 shows the mapping fields web page 520 for selection of pdf settings 5210 with an instruction to preview before submit 2830 of one embodiment.

#### PDF Block Condition Mapping Fields all Elements:

[0260] FIG. 53 shows for illustrative purposes only an example of pdf block condition mapping fields all elements of one embodiment. FIG. 53 shows mapping fields 2810 where a user can select an all element drop down 5310 and select a field including name 5315 of one embodiment.

#### PDF Block Condition Mapping Fields Table:

[0261] FIG. 54 shows for illustrative purposes only an example of pdf block condition mapping fields table of one embodiment. FIG. 54 shows the mappings field 5400 webpage and an insert 5410 feature selection menu that includes a table 5425 with a menu that includes insertion of a page break 5420. Showing is a table grid 5430 where the user highlights a 3x1 page break 5435 for the table of one embodiment.

#### PDF Block Condition Starting Mapping Fields:

[0262] FIG. 55 shows for illustrative purposes only an example of pdf block condition starting mapping fields of one embodiment. FIG. 55 shows mapping fields 2810 with an instruction click here and open sub menu with right click 5510. A sub menu 5520 that includes an insert drop down 5530 and shows a user selected name 5540 that the user can apply 5550 of one embodiment.

#### PDF Block Condition Mapping Field Name:

[0263] FIG. 56 shows for illustrative purposes only an example of pdf block condition mapping field name of one embodiment. FIG. 56 shows the mapping fields 2810 where a name 5315 is displayed and a name insert box 5620 can be entered or automatically inserted from a platform account field of one embodiment.

#### PDF Block Condition Mapping Fields:

[0264] FIG. 57 shows for illustrative purposes only an example of pdf block condition mapping fields of one embodiment. FIG. 57 shows the mapping fields 2810 where a name entry displays the name 5315, phone 1260 and email 1070 associated with the name entered of one embodiment.

#### PDF Block Condition Mapping Fields Selection:

[0265] FIG. 58 shows for illustrative purposes only an example of pdf block condition mapping field selections of one embodiment. FIG. 58 shows a mapping field web page 5800 that a user has setup with a greeting hello 5850, a name 5315 and a salutation welcome to my dynamic pdf 5854. Dynamic mode is on 5810 and allow on 5820 indicates it will be activated as directed in an entry in a condition drop down 5830 selection from the condition drop down sub menu selection table row table col 5840 of one embodiment.

#### PDF Block Rule Condition Selection:

[0266] FIG. 59 shows for illustrative purposes only an example of pdf block rule condition selection of one embodiment. FIG. 59 shows a rule condition selection 5900 with a name drop down 5910 and condition drop down rules for example equals 5920 john 5930 of one embodiment.

#### PDF Block Condition Mapping Fields Dynamic Mode Settings:

[0267] FIG. 60 shows for illustrative purposes only an example of pdf block condition mapping field dynamic mode settings of one embodiment. FIG. 60 shows mapping fields 2810 with dynamic mode on 5810 and allow on 5820 where the greeting and salutation includes a name 5315, phone 1260, and email 1070 of one embodiment.

[0268] PDF Block condition. The ‘Block condition’ is a conditional logic that can be used to show and hide selected parts of the PDF. This option can only be used in the Dynamic PDF mode. The block condition can allow a user to set 3 types of conditions:—Selection—to show selected parts in the PDF—Table Col—to show a table column—Table Row—to show a table row. Following is a step-by-step example for all 3 types of block condition. Create a new simple form. Drag 3 elements: 2 textboxes and an email field—Select each field on canvas and go to the ‘Properties’ panel>‘Element’ tab>‘Settings’ option>‘Basic’ category>Label and change their label names to: Name, phone and email. Create a manual PDF:—Go to ‘Properties’ panel>‘Form’ tab>‘Settings’ option>Press “view more” link>‘PDF Mapping’ category. Press on the “Map fields” button and the PDF editor will then open. Use the toggle to change to the Dynamic mode. Press on the “Settings” icon in the Toolbar and turn on the “Preview before submit” checkbox (in the General category). Add some text, for example “Hello” and add the “name field” right after it. So it will now say: Hello name, Add some more text under that, for example “Welcome to my dynamic pdf”. Add a table with 3 columns (toolbar>insert>table). Click inside the first column and open a sub menu with a right mouse click. Select Insert and choose the ‘Name’ field—it will then be inserted into the column. Move to the second column and insert the ‘Phone’ field into it. Move to the third column and insert the ‘email’ field into it. At this point in the example a user can create 3 different types of Block conditions: Selection, Table Row, and Table Col.

[0269] Following is an example for each type: Selection—Hide/show selected part of the PDF. Select the text added at the top (“Hello name . . .”), right click to open the sub menu. Select ‘Condition’ and then ‘Selection’. Now a window will open for the user to configure the condition: this example is a very basic conditional window where the result is always the same and if the condition is met, then the selected item/s

will show. Now set the following condition: Name equals John and press “apply”. This means that if John will be entered in the name field—the text the user selected will appear. The user will notice that the selected area is now confined within an “If Block” box. Clicking on the ‘If block’ title tag will open the condition window for edit. Press ‘Apply’ and save the form. Test the PDF: Publish the form as URL. Enter the form: John in the ‘name’ field, 1234 in the ‘phone’ field and ‘john@test.com’ in the email field, then press on submit. The PDF preview will open and the text in the top part of the PDF will appear (because the condition was met). Now, press cancel and change the name to “Jill”.

[0270] Open the preview again and the user will not see the text at the beginning of the PDF. Table Col—Hide/show table column—Click inside the third column (with the email in it).—Right click to open the sub menu.—Select ‘condition’ and then “table col”. Now configure the condition: phone contains 1 this means that if the form filler enters a phone number that contains the digit 1 in it—the email cell will show in the PDF table. Press ‘Apply’. The user will now notice that the selected column content is now confined within an “If Block” box. Clicking on the ‘If block’ title tag will allow the user to edit or remove the condition. Press ‘Apply’ and save the form. Test the PDF: Publish the form as URL. Enter John, 1234 and john@test.com—Press on submit—The PDF preview will then open and the table will show all 3 columns. Now press ‘cancel’ and remove the digit 1 in the phone number. Open the PDF preview again and the user will not see the third column (with the email).

[0271] Table Row—Hide/show table row—Click inside the second column (with the phone in it)—Right click to open the sub menu—Select ‘condition’ and then “table Row”—Now configure the condition: Name is not empty This means that if the form filler enters any name then the table row will show. Press ‘Apply’. The user will also notice that the selected row content is now confined within an “If Block” box. Clicking on the ‘If block’ title tag will allow the user to edit or remove the condition. Press ‘Apply’ and save form. Test the PDF: Publish the form as URL. Enter John, 1234 and john@test.com. Press on submit and the PDF preview will then open and the table row will appear. Now press ‘cancel’ and remove the value from the Name field. Click submit again and the user will not see the table row of one embodiment.

#### A Sentiment Analyze Feature:

[0272] FIG. 61 shows for illustrative purposes only an example of a sentiment analyze feature of one embodiment. FIG. 61 shows a sentiment analysis 3360 feature with a sentiment enlarged view 6110 of results from a form: contact us 6120. The form: contact us 6120 results include sentiments expressed including negative 76.1%, positive 78.5%, and positive 78.5% 6130 of one embodiment.

[0273] FormTitan launches the first Form Builder employing sentiment analysis for understanding textual responses. Focusing on conversion rate optimization (CRO). The FormTitan team realized that people need more than a tool to build good looking forms, they want to improve conversion rates. So they built an engine that goes over every newly created form, analyzes it, and provides a user with suggestions for changes which would improve conversion rates. If the form has open-ended text fields, FormTitan can employ Sentiment Analysis algorithms to analyze the data. So, for example, it can highlight customers at risk and even use its

built-in conditional logic to notify those in charge of customer retention or show a predefined message to such users.

#### Heat Map Integration:

[0274] FIG. 62 shows for illustrative purposes only an example of heat map integration of one embodiment. FIG. 62 shows a heat map **6200** displayed. The heat map displays the areas of the form most frequently scanned by visitors **6230** to indicate traffic **6210** concentrations. A color scale showing a low to high heat level corresponding to the form area most frequently scanned by visitors in a graphic format **6220** of one embodiment.

[0275] Use Heatmap to optimize a form. One of the many optimization tools in FormTitan is the Heatmap. This feature provides a user with a graphical representation of how user form visitors move along the form. It colors the more “active” parts of the form in warmer colors (yellow, orange and red) and shows the areas where the users don’t go through as much in cold colors (blue and green). The heatmap draws a kind of map, where a user can see where form fillers spend more time. It helps a user understand which parts of a user form are the strongest and which are the weakest and place the elements in the user form accordingly.

[0276] Following are steps to use the heatmap feature:  
 1—Open a form in the form builder. 2—Go to “Properties” panel>“Form” tab>“Settings”>“Optimization” category.  
 3—Turn on the “Heatmap” checkbox and save the form.  
 4—A user can also turn this feature on from the “My Forms” page, by choosing the form on the left and using the toggle in the “form optimization” section to turn it on. 5—Once the feature is turned on; all a user has to do is wait. When form fillers start to use the user form and fill in data, the form fillers movements will be recorded and the user will be able to see the resulting map of one embodiment.

#### Integrating a Form Platform into Form Builder:

[0277] FIG. 63A shows for illustrative purposes only an example of integrating a form platform into form builder of one embodiment. FIG. 63A shows the form **4210** settings [?] **4220** for integrations **6302** including push notification **6303**, set notification **4230** and map fields **4250**. The map fields **4250** include contact us **3726**, user name **6311**, description **6312**, and submit **6313** of one embodiment.

[0278] Form Builder can integrate multiple applications. These are a variety of calendar/schedulers that can be integrated into a responsive form using FormTitan. A user can create a form to integrate with a calendar application that is responsive by manual entry and voice commands.

#### Authenticate a Form Platform:

[0279] FIG. 63B shows for illustrative purposes only an example of authenticate a form platform of one embodiment. FIG. 63B shows using push data **2210** into a selection made using choose platform **6322** for example Outlook Calendar **6330**.

[0280] An event calendar feature is illustrated with a Salesforce Event Object. Integration with the Salesforce Event Object FormTitan is an online web tool that allows a user to create online forms, landing pages and Salesforce forms. The FormTitan form builder can easily integrate with Salesforce event object. Not only does this require no coding skills on a user’s part, but the Salesforce connector is seamless and powerful. The Salesforce connector not only

writes to a Salesforce event object but it can also read from it and populate user online form fields instantly. Furthermore, a user can read from multiple objects at a time, with or without dependencies and query objects based on any value from the user form. An Event Object represents an event in the calendar. In the user interface, event and task records are collectively referred to as activities. Usage Use Event to manage calendar appointments.

#### Choosing Form Fields Data to Push to Form Platform:

[0281] FIG. 63C shows for illustrative purposes only an example of choosing form fields data to push to form platform of one embodiment. FIG. 63C shows push data **2210** for creating platform parameters **6340** in this example for Outlook Calendar **6330**. One platform parameter is 1. Authenticate with Microsoft: authenticated **6350**. Then the 2. fill Outlook Calendar parameters **6351** selections include email **1070**, end datetime **6353**, description **6312**, start datetime **6355**, summary **6356**, and finish **6370**. Displayed is a notification and instruction that push is inactive—turn notification “on” to activate **6360**. A notification **6361** area shows a notification activation switch shown in off **6362** position. A user will turn on the notification toggle and press on “Finish”. Once the form is configured to integrate with Outlook Calendar—a user will fill in a form and then a new appointment will be added in the user calendar based on the user’s form push configuration of one embodiment.

#### An Event Calendar Feature:

[0282] FIG. 64 shows for illustrative purposes only an example of an event calendar feature of one embodiment. FIG. 64 shows Outlook Calendar event listings **6400** of one embodiment.

#### A Publish Form Feature:

[0283] FIG. 65 shows for illustrative purposes only an example of a publish form feature of one embodiment. FIG. 65 shows a publish form **6500** feature. A user can choose publish type **6510** including</> embed **6520** and copy paste **6530**. The area for a publish form section includes user settings for width: for example **1200 6532** and height: for example **900 6534**. Other publish features include publish option: Android APK, and Supporting URL parameters in publish options: Embed, Lightbox and Feedback of one embodiment.

[0284] Publish Form features have several options for publishing a user form. Once a user is happy with a form, click on the publish button in the upper toolbar. A publish window will then open with all the publishing options available. A user can select an option and a generated code will be created accordingly. Copy the code and paste it in the user website, email, blog etc. In addition to all the publish options, the publish window first provides the user with the form URL. Clicking on “view” will open the URL in a new browser tab, enabling a user to test it. Following is a list of publishing options in FormTitan: In each of the options a user can select and modify the properties below (width/height, link text, border, time, target etc.). 1—Embed: Use this option to embed the form within a user web page. 2—iFrame: Use this option to embed the form within the web page in an inline frame. 3—LightBox: Use this option to generate a link that opens the form on the page as lightbox. 4—Feedback: This option adds a button to the

bottom right corner of the web page. When pressed—the form opens. A user may also choose one of the 4 button styles available. 5—Link: Use this option to generate a direct link to the user selected form. 6—PopUp: This option generates a link that when pressed it opens a new pop up window. 7—Email: Allows a user to send an email containing a link to the selected form. A user can enter a recipient, subject and email body. 8—WordPress: Generate a code that can be embedded in a user application platform. 9—QR Code: Automatically generate a QR code pointing to the form. 10—Image: This option generates a QR code image of the form, that when clicked it leads to it. 11—Exit Form: This generated code, when inserted in the user web page, makes sure that the form will appear if/when the user tries to leave the page. 12—Timed Form: Lets a user set a timer (in seconds) for when a user would like to make the form appear. 13—HTML File: Generates an HTML code a user can use. 14—Targeting: Lets a user send the form to a mailing list or phone list. 15—Application platform codes feature that generates a code a user can use in application platform sites. A user can publish the form using the following instructions: 1—Enter the form builder and open the form to use. 2—Once the form is on a canvas then press on the “Publish” icon. 3—Select a publish type and copy the code below into the publish type box of one embodiment.

A Voice Interactive Calendar and Geo Locator:

[0285] FIG. 66 shows for illustrative purposes only an example of a voice interactive calendar and geo locator of one embodiment. FIG. 66 shows bi-directional voice command and interactive information from calendar of events and geo location mapper 6600. The user smart phone 3710 with the FormTitan application installed can display a mapper with information overlays including for example apt: 1:45 pm, ETA: 2:66 pm, accident with heavy traffic delays ahead 6610. A FormTitan network digital processor can calculate the remaining distance to the user's next appointment and estimate arrival time based on projected travel speeds using current traffic conditions. The user smart phone 3710 displays calendar and geo locator 6625. The FormTitan application transmits an audible message “late to next apt—call or text ETA?” 6630 to the user 3730 who may be driving. The user using a hands-free ear piece hears the FormTitan application audible message 3880. The user instructs the application to “text apology and ETA due to traffic accident delays” 6650. The FormTitan application sends a text to the next appoint location 6655 sending the user instructed message of one embodiment.

[0286] The bi-directional voice command and interactive information from calendar of events and geo location mapper can greatly assist a user in making their appointments. In this example the voice element notifies the user while driving that his next Apt: 1:45 PM, ETA: 2:10 PM accident with heavy traffic delays ahead from the user's event calendar. On the screen the geo locator shows where his next appointment is and the traffic status acquired from local law enforcement notifications and a calculated ETA based on the distance and reduced speeds.

[0287] The voice element asks “late to next apt—call or text eta?” The user instructs the customized customer relationship management platform through the voice element to “text apology and eta due to traffic accident delays”. The interactive capacity of the customized customer relationship management platform has allowed the user to get valuable

information and assist the user in providing a professional image with his contacts of one embodiment.

Salesforce Lookup:

[0288] FIG. 67 shows for illustrative purposes only an example of Salesforce lookup of one embodiment. FIG. 67 shows a Salesforce lookup 6700 menu selection lookup 6710 the user can press which opens a lookup text box 6720. The user can enter the data into the lookup text box 6730. Another feature includes Date and Date time format in Table and Lookup of one embodiment.

[0289] The Lookup is a special element that is used to integrate forms with Salesforce. It allows a user to create a search box for users, enabling them to search inside Salesforce objects and choose an item. To add a Lookup, a user can set the integration with Salesforce and configure the columns of the lookup results table. Following are steps to use the Lookup element: 1—Enter a form in the form builder. 2—Go to “Elements” panel>“Salesforce” category Drag the “Lookup” element onto the canvas. 3—Once the “Lookup” is selected on canvas its properties will open in the panel on the right. In “Properties” panel>“Element” tab>“Settings” option>“Basic” category>Mapping) press on the “Set” button. 4—A window will now open, to configure the Lookup: Authentication First, authenticate with Salesforce. Configure Salesforce integration. Press on the “Configure Salesforce Integration” button to make the connection between the Lookup and the Salesforce object. This window looks very much like the regular integration mapping window, only it is designed for the lookup and contains only 4 categories: Connection—this will contain the account a user authenticated, a user can change it. Salesforce object settings—Select the Salesforce object the user want form filler users to search in. Condition: Set a condition for example Account Name Contains Search box (which means the user will enter text in the searchbox and matching items will be searched for in the “Account Name” field). Sort order: This is not a mandatory, however a user can control how the results will be sorted (according to which fields) and if the sorting is Ascending or Descending. Configuring the results table columns. The results of the search will be displayed in a table—a user will be required to set the columns of this table. Open the “Add fields for display” dropdown, choose the fields, one by one, and press “Add”. Each field will be displayed in a row and will include: Col—This is the field name Custom label—an input box in which a user can enter a different text for the field label (instead of the one specified in the Col). Text & Value includes 2 sets of vertical radio buttons that control which field will be used for the Text in the results, and which will be used to store the value. Press “Apply” and save the form. Now a user will be able to publish the form as URL and test it. 5—Testing When the form loads a user can either: A—Enter the text inside the Lookup element and press on the “Search” button. The results will then open in a window according to a user search. B—Press on the Lookup button to open it, enter a text in the searchbox and press on the “Search” button. The results will then appear according to the user's search.

[0290] Mapping the Salesforce External Lookup Relationship field to a user form field. The Salesforce External Lookup Relationship field type is an external lookup relationship links a child standard, custom, or external object to a parent external object. A user can map an External Lookup

Relationship field from Salesforce to a user FormTitan form field. A lookup relationship creates a relationship between two records so a user can associate them with each other.

#### Advanced Table:

**[0291]** FIG. 68 shows for illustrative purposes only an example of advanced table of one embodiment. FIG. 68 shows an advanced table **6800** feature to create a table **5425**. A table selection opens a table formatting template **6820** for a user to select a column for example col 1 **6830** and select a number of rows for example 4 rows **6840** of one embodiment.

**[0292]** Salesforce Table Working with Salesforce in an organized way. The Salesforce Table is an element a user can drag from our “Elements Panel” and use in the user form, but it may only be used for working with Salesforce. Since FormTitan allows a Bi-Directional integration with Salesforce, a user can draw data from Salesforce to a user table, and edit that data so it is updated in the Salesforce object. A user can read & write using a table element.

**[0293]** Simple Table. Since Salesforce works with objects, the table element is built according to it. Once a user drags and drops the table element the user will set its columns and then configure the Salesforce “Get” integration to populate them. This is the basic use of the table.

**[0294]** Advanced Table. FormTitan Salesforce form builder allows a user to create more complex scenarios. A user can make the table hierarchical, drawing data with parent-child relationships and also writing to different objects simultaneously.

**[0295]** Simple Table. When using the Table element to create Salesforce forms, a user can use the simple table or the advanced table. The “simple table” refers to the most basic table a user can create for working with SF. It is not hierarchical—It has a single level, with which a user can only “Read” data from SF, and a user can do so from only one Salesforce object. In order to configure a simple table all a user will need is to change the table’s “Basic” settings and set the “Get” integration. The “Advanced table”, in comparison, is more powerful. It refers to a hierarchical table that allows a user to work with multiple Salesforce objects and allows a user to “Read” from Salesforce objects as well as “write” to Salesforce objects.

**[0296]** Creating a simple table is easy—just follow these steps: 1—Enter a new/a user form in the form builder 2—Drag a table element from “Elements” panel>Widgets>Table 3—Enlarge the table on canvas using the resizing handles 4Once selected, the table’s properties will open in the “Properties” panel (on the right side) 5—Change the default column values from Col1, Col2, Col3 to the Salesforce object fields a user would like to display in the table. For example: if a user wants to display information from the “Account” Salesforce object a user could set the following columns: Account Name, Account Phone, Account Website. \*

**[0297]** Please notice that this table is “Flat”, it has no hierarchy and its columns can belong to only one Salesforce object. 6—A user can, of course, add more columns, or remove columns according to your requirements. 7—A user can add a filter row to the table, just below the headers, to help the user search for specific data—by turning on the “Show filter checkbox. 8—Change the table style—the table’s appearance, just like any FormTitan element, can be easily altered in the Element style. Go to: “Properties”

panel>“Element” tab>Style” option>and open the “Apply style to” drop down. Choose the part of the table a user would like to apply a style to. Weather it is to the table in general, to the rows, header etc. 9—make the style changes in the relevant categories. For example: Choose “Table” in the “Apply style to” dropdown, and specify a pale blue color in the background. 10—All that is left to do now is to connect this table to the Salesforce object. Go to the “Properties” panel>“Form” tab>“Settings”>open the “Salesforce Integration” category

**[0298]** Notice that there are two sections here: the top one is “Push to Salesforce”, in which a user can configure the “writing” operations (creating, updating, upserting, deleting data in Salesforce objects). And the bottom section is “Get from Salesforce”, in which a user can configure the “Reading” from Salesforce objects. Choose the in the “Get from Salesforce” section and press on the “Map Fields” button. 11—Authenticate with Salesforce and press on the “Add object” button 12—Choose the Salesforce object a user would like to “read” from. In this case: Account. 13—Turn on the “execute on form load” checkbox—this will “pull” the data from Salesforce and populate the table once the form is loading. 14—There is no need to set a condition in this case, however, a default will display “All” matches found. As well as limit the number of accounts shown to 200. 15—Map the form fields (in this case, a user table columns) to the Salesforce object fields. And press on “Apply”. 16—A user will see that a single integration line has been created. (A user can edit it using the edit icon on the right). Press on “Apply” and save the form. 17—Publish a user form to see the result. Press on the “Publish” icon in the main toolbar above Press on the “View” button to see the form. 18—And this is the published form—with the data populated straight from Salesforce—in real-time.

**[0299]** Advanced Table. When creating online forms for Salesforce with the Table element a user can create a simple table, that only allows to read data from Salesforce, work with in a flat mode (with no hierarchy) and connect to only one Salesforce object, FormTitan still provides the capability to create a more powerful table which allows a user to: 1—Read from Salesforce objects as well as write to Salesforce objects. 2—Work with many Salesforce object at once. 3—Work with hierarchy—data that has parent-child relationships. Create as many levels as a user wants in the table. A table is considered advanced once a user can start adding to the “simple” table—adding levels, adding Salesforce objects, and adding “writing to Salesforce objects.

**[0300]** Creating an advanced table is done in 3 phases: First Phase: Creating the table, it’s levels and columns Second Phase: Setting up the Salesforce integration Third Phase: Setting form filler permissions to View, Edit, Add, Delete, Export data First Phase: Creating the table, it’s levels and columns 1—Enter a new/user form in the form builder 2—Drag a table element from “Elements” panel>Widgets>Table 3—Enlarge the table on canvas using the resizing handles. 4—Once selected, the table’s properties will open in the “Properties” panel (on the right side). 5—Change the default column values from Col1, Col2, Col3 to the Salesforce object fields a user would like to display in the table. For example: if a user wants to display information from the “Account” Salesforce object a user could set the following columns: Account Name, Account Phone, Account Website.

[0301] This is the configuration for the first level of the table—displaying the parent object data. 6—A user can add more columns, or remove columns according to a user's requirements. 7—A user can add a filter row to the table, just below the headers, to help the user search for specific data—by turning on the “Show filter” checkbox. 8—Max Rows property allows a user to enter a maximum number of rows that can be displayed in the table level. If, for example a user can set a max number of 10, then the form filler will only be able to add rows to this Table level until it reaches the max number. After that—the “Add” button will become disabled. 9—The ‘Rows per page’ property allows a user to define how many rows will the table display in each page. 10—Change the table style—the table's appearance, just like any FormTitan element, can be easily altered in the Element style.

[0302] Go to: “Properties” panel>“Element” tab>Style” option>and open the “Apply style to” drop down. Choose the parts of the table a user would like to apply a style to one by one. A user can change the style of the table in general, the rows, table header, columns and even the modal window of the “Edit”/“Add”/“View”/“Delete” options. 11—Make the style changes in the relevant categories. For example: Choose “Table” in the “Apply style to” dropdown, and specify a pale blue color in the background. 12—Ok, so now that the first level is configured a user can start creating the next level in the table. Go to “Properties” panel>“Form” tab>“Settings” options>“Advanced” category press on the “Add level” button. A new level will appear on canvas containing the default columns (Col 1, Col 2, Col3) In addition 2 links will now be added inside the table element on canvas, for moving between the levels and also a “Previous” button for the user—to go back to the first level. 13—Set the fields of the child Salesforce object. “Properties” panel>“Element”>“Settings” option>“Basic” category>change column names.

[0303] For example: if a user wants to display the Contacts of each account a user could set the following columns: Contact last name, Contact first name, Birth date. Second Phase: Setting up the Salesforce integration. 14—Start by going to the Salesforce integration. Go to the “Properties” panel>“Form” tab>“Settings”>open the “Salesforce Integration” category. Notice that there are two sections here: the top one is “Push to Salesforce”, in which a user can configure the “writing” operations (creating, updating, and upserting, deleting data in Salesforce objects). And the bottom section is “Get from Salesforce”, in which a user can configure the “Reading” from Salesforce objects. Choose the “Get from Salesforce” section and press on the “Map Fields” button. 15—Authenticate with Salesforce and press on the “Add object” button 16—Start mapping according to user table levels—start from the first level and work down.

[0304] So the first Salesforce object a user would need to “read” from. According to this example is Account. Open the dropdown and choose it. 17—Turn on the “execute on form load” checkbox—this will “pull” the data from Salesforce and populate the table once the form is loading. 18—There is no need to set a condition in this case, however, the feature will display “All” matches found. As well as limit the number of accounts to show to 200. 19—Map the form fields (in this case, level #1 table columns) to the Salesforce object fields. And press on “Apply” 20—Once done, a user will see that a single integration line has been created. This line, holds mapping to the first level

of the table. (A user can edit it using the edit icon on the right). Press on “Apply” and save the form. 21—Now a user will need to add the integration for the second level of a table and map the contact fields.

[0305] To display the contacts belonging to each account in hierarchy, a user will need to add the next object as a child of the first object. Press on the “Add object” button located at the bottom of the window. 22—Select the “Contact” object from the drop down. A user will now need to add a condition in order to set the parent-child relationship like so: Choose to display “All” matches found. As well as limit the number of accounts to show to 200. 23—And map user contact fields to the Contact object fields in SF. press “Apply”. 24—A user will now see 2 integration lines. The first is the account line, and the second line, will hold the mapping to the contact object. 25—Publish a user form to see the result. Press on the “Publish” icon in the main toolbar above. Press on the “View” button to see the form. 26—And this is the published form—with the data populated straight from Salesforce—in real-time. On the left part of the table a user will see the columns a user sets.

[0306] On the right part will see a column called “Next level”, which holds buttons that will lead to the data in the second level. Third Phase: Setting form filler permissions to View, Edit, Add, Delete, Export data 27—Making table columns editable Now that table levels are set, and the Salesforce integration configured by a user can allow a user form filler to “Read” data from Salesforce. If a user wants to add more functionality and allow “writing” a user can easily do that by setting permissions. “Properties” panel>“Form” tab>“Settings” options>“Advanced” category and turn on the checkboxes according to user needs:—Allow view—this will allow the form filler to view more data than is displayed in the table columns. When a user turns on this checkbox a configure button will appear and a user will need to press on it. A window will then open listing this level’s column names and a user will need to turn on the checkboxes of the fields a user wants to make viewable. In addition a user will be able to add fields for viewing.—Allow edit—this will allow the form filler to edit the data in this level and by doing so, to update the Salesforce object when a user turns on this checkbox a configure button will appear and a user will need to press on it. A window will then open listing this level’s column names and a user will need to turn on the checkboxes of the fields a user wants to make editable.

[0307] Allow add—this will allow the form filler to add new data in this level and by doing so, to add data inside the Salesforce object. When a user turns on this checkbox a configure button will appear and a user will need to press on it. A window will then open listing this level’s column names and a user will need to turn on the checkboxes of the fields a user wants have the user fill in when the user wants to create a new item. A user can restrict the number of rows in the table by mapping the “MaxRows” property of the table in the GET operation. If a user can map that then the user can only add rows as long as the total rows in the table are less than the MaxRows set to. —Allow delete—this will allow the form filler to delete data in this level and by doing so, to have the data deleted in the Salesforce object.

[0308] Allow Export—this option will allow the form filler to export the data in the current level he is in. When a user turns on this checkbox a new “Export” button will be added at the bottom of the table and when the user presses

on it the data in the level will be downloaded in CSV. 28—if a user wants a form filler to be able to update the second level of the table can set the “writing” permissions, just like a user did in the first level. (go to “Properties” panel>“Form” tab>“Settings” options>“Advanced” category, and turn on the checkboxes according to user needs) 29—Adding fields to the View/Edit/Add windows (in addition to the column fields) While a user wants a user table to stay compact and show only the most important fields, a user may want to have fields added to the Add/Edit modal windows so they may also be updated. Adding a field is easy—all a user has to do is choose it from the dropdown below and press on “Add”.

[0309] Once a user adds a field to the Add/Edit window a user will have 3 options: 1—The field will be added to the specific window and its data loaded from Salesforce. Once a user adds a field its default mode will be to load the data from Salesforce and a user will indeed see that the “Load data” checkbox is turned on. 2—The field will be added to the specific window and its data will be loaded from a selected field in a user form. In order to do this a user will need to turn on the “Map value” checkbox (this will remove the “load data” checkbox).

[0310] A dropdown will then appear on the right containing fields from the form. Read more about loading a selected field from Salesforce 3—The field will be added to the specific window and its data will be loaded from a field within a user Table. In order to do this a user will need to turn on the “Map value” checkbox. A dropdown will then appear on the right containing the object in Level 1, and once selected another dropdown will appear so a user can select the field a user wants to map to. Read more about loading the field from a user’s Salesforce Table 30—if a user has a hidden field in a user form, will be able to store the relevant record ID in it—the record ID of the viewed record, the edited record or the added record (depending on which window a user is doing this from). Read more about Map record ID to hidden 31—Now go and test a user form again: If a user has added the permission to edit or delete the items in the table—a user will also see 2 more columns, containing links to edit & delete. If a user has added the permission to add a new item—an “Add” button will be added at the bottom of the table. Populating user Salesforce data in a Table element. Mapping the Salesforce fields to a user FormTitan Table element. Populating data from Salesforce inside a user online form is easy—all a user has to do is use the integration and map the Sf fields to the fields in a user form. A user can display the data in a table. FormTitan has a table element especially for working with Salesforce. Following are the steps to draw Salesforce data into a user form. FormTitan table element: 1—Drag a table element (“Elements panel”>“Widgets” category>“Table”). 2—Configure the column headers (In the “Properties panel”>“Element”>“Settings”). 3—Change the table size and styles (In the “Properties panel”>“Element”>“Style”). 4—at this point a user can also hide the table filter and decide how many rows a user wants to be displayed per page. 5—Integrate with Salesforce to draw data (In the “Properties panel”>“Form”>“Settings”>“Salesforce Integration” category>“Get”). 6—Push on the “Get” button and authenticate with Salesforce. 7—Choose the “account” object. 8—Turn on the “Execute on form load” checkbox. 9—Choose the “All” option in the “If multiple matches found take ID from” field, and give a limit number (in this case: 300). 10—Map the fields and press “Apply”. 11—An

integration “Get” line will be created. 12—Press on “Finish” and save the form. 13—Publish a user form as URL to test it. A user will see the user Salesforce accounts are displayed inside the table when the page loads. Populating into FormTitan elements. Salesforce data can also be populated into other form elements such as: Dropdowns or radio buttons (for Picklists), Textboxes, Text area, paragraph (for Text area rich), Date, repeatable sections etc. Data can be populated on form load, but it can also be dynamically populated upon a trigger.

Mapping the Salesforce Fields to a User FormTitan Table Element:

[0311] FIG. 69 shows for illustrative purposes only an example of mapping the Salesforce fields to a user FormTitan table element of one embodiment. FIG. 69 shows mapping the Salesforce fields to a user FormTitan table element 6900. The user can use an untitled form to select an account name 6910, phone 1260 and website 6930. FIG. 69 shows below a corresponding reference list of accounts matching the selected account name 6910, phone 1260 and website 6930 of one embodiment.

[0312] Populating user Salesforce data into a Table with hierarchy. Mapping the Salesforce fields to a user FormTitan hierarchical Table element. FormTitan makes it possible to populate your Salesforce data in a table element. The Table element can be made hierarchical by adding additional levels to it. A grid is initially displayed to a user and shows the results of the first level, for example: accounts details. The user can then decide to focus on a certain row in the grid and drill down to view the second level (by pressing on an arrow link of the relevant row). For example: The user is shown a list of accounts.

[0313] A user picks one and the user is then shown all the contacts belonging to this account. Following are the steps to draw Salesforce data into a user FormTitan hierarchical table element: 1—Drag a table element (“Elements panel”>“Widgets” category>“Table”). 2—Configure the column headers (In the “Properties panel”>“Element”>“Settings”). 3—Change the table size and styles (In the “Properties panel”>“Element”>“Style”). 4—A user can also hide the table filter and decide how many rows a user wants to be displayed per page. 5—Now a user will need to add another level. In the “Properties panel”>“Element”>“Settings”>Advanced>press on “Add level” button. Once the button is pressed a user will see that the previous level columns will be replaced on a canvas by the new level’s default columns (Col 1, Col 2, and Col 3) and a user will need to configure the new level’s columns: Contact Last Name and Contact First Name. (Delete the third column).

[0314] Please notice that Level links are added on a canvas, to the bottom of the grid on the left. These links will allow a user to move between the table’s levels on canvas. 6—Integrate with Salesforce to draw data into the grid (In the “Properties panel”>“Form”>“Settings”>“Salesforce Integration” category>“Get”). 7—Press on the “Get” button and authenticate with Salesforce. 8—Choose the “account” object. 9—Turn on the “Execute on form load” checkbox. (This will draw the accounts once the form is loaded). 10—Choose the “All” option in the “If multiple matches found take ID from” field and enter a limit number (in this case: 300). 11—Map the fields (account name, phone and website) and press “Apply”. 12—A “Get” integration line

will be created. 13—Press on the “Add object” button and choose the “Contact” object. (This is the object a user wants to read from in order to populate fields in level 2 of the grid). 14—Set a condition: Account ID Equals fld2 Level 1 RecordID of one embodiment.

**[0315]** Choose the “All” option in the “If multiple matches found take ID from” field. And enter a limit number (in this case: 300). 15—Map the fields: Last name and First name and press “Apply”. 16—Another “Get” integration line will be created and there will be two. 17—Press on “Apply” and save the form. 18—Publish a user form as URL to test it. A user will see the user Salesforce accounts are displayed inside the table when the page loads—this is the first level. 19—Press on the “Arrow” link in a certain account row to drill down and see its contacts. Populating into FormTitan elements Salesforce data can also be populated into other form elements such as: Dropdowns or radio buttons (for Picklists), Textboxes, Text area, paragraph (for Text area rich), Date, repeatable sections etc. Data can be populated on form load, but it can also be dynamically populated upon a trigger.

**[0316]** Changing the Salesforce Table Styles When a user wants to add a Salesforce Table to a user form, all a user needs to do is drag a single element called “SF Table” onto the canvas. However it is important to understand that the Table has a few parts in terms of style (like header, rows, modal windows etc.) and so a user needs to choose which part to apply the style to. After a user chooses a part a user can easily change its style using our regular style categories: Font, Background, Size & Position, Margin, Padding, Alignment, order and CSS Name. Following are examples of how to change the style of each part: 1—Enter a user form in the form builder. 2—Drag a Table element from the “Elements” panel>“Salesforce” category. 3—When the Table is selected Go to “Properties Panel”>“Element” tab>“Style” Option. Table 4—Open the “Apply style to” dropdown and choose the “Table” option. This option allows a user to change the style of the entire table in general changing font, background, size, etc. will affect the Table as a whole. Row 4—Open the “Apply style to” dropdown and choose the “Row” option. This option allows a user to change the style of the all the grid rows Style changes will effect only the table rows.

**[0317]** Alternative row 4—Open the “Apply style to” dropdown and choose the “Alternative Row” option. This option will allow a user to have alternating row styles. It will affect every second row in the grid. Header 4—Open the “Apply style to” dropdown and choose the “Header” option. This option allows a user to change the style of the table header. Add Button 4—Open the “Apply style to” dropdown and Choose the “Add Button” option. This option will allow a user to change the style of the Add Button, that appears at the bottom of the Table. A user can change its size, colors, border, padding alignment etc. Export Button 4—Open the “Apply style to” dropdown and choose the “Export Button” option this option will allow a user to change the style of the Export Button that appears at the bottom of the Table.

**[0318]** A user can change its size, colors, border, padding alignment etc. Next Button 4—Open the “Apply style to” dropdown and choose the “Next Button” option. This option will allow a user to change the style of the Next Button, that appears inside the Table, in its own column. A user can change its size, colors, border, padding alignment etc. Previous Button 4—Open the “Apply style to” dropdown and

choose the “Previous Button” option. This option will allow a user to change the style of the Previous Button that appears at the bottom of the Table in a child level. A user can change its size, colors, border, padding alignment etc.

**[0319]** Modal Header 4—Open the “Apply style to” dropdown and Choose the “Modal Header” option. If a user allows a user form filler to update data through the table: View, Edit, Delete and Add data, all this will be done in modal windows. A user can change the windows style of these 4 modal windows. This option will allow a user to change the style of the Modal window title Modal body 4—Open the “Apply style to” dropdown and choose the “Modal Body” option. If a user allows a user form filler to update data through the table: View, Edit, Delete and Add data, all this will be done in modal windows. A user can change the windows style of these 4 modal windows.

**[0320]** This option will allow a user to change the style of the Modal window title (in the 4 modal windows) Modal button Apply 4—Open the “Apply style to” dropdown and Choose the “Modal Button Apply” option. If a user allows a user form filler to update data through the table: View, Edit, Delete and Add data, all this will be done in modal windows. A user can change the windows style of these 4 modal windows. This option will allow a user to change the style of the Modal Apply Button (in the 4 modal windows). Modal button Cancel 4—Open the “Apply style to” dropdown and choose the “Modal Button Cancel” option. If a user allows a user form filler to update data through the table: View, Edit, Delete and Add data, all this will be done in modal windows.

**[0321]** A user can change the windows style of these 4 modal windows. This option will allow a user to change the style of the Modal Cancel Button (in the 4 modal windows). Col 4—Open the “Apply style to” dropdown and Choose the “Col” name option (in our example we did not change the default names of the columns and by default columns are named “Col1”, “Col2”, “Col3”, however after a user changes column names they will be the ones listed in the “Apply style to” dropdown). This option allows a user to change the style of a column in the table: its color, its width etc. Next Level 4—Open the “Apply style to” dropdown and choose the “Next Level” option. This option allows a user to change the style of the Next Level column in the table.

**[0322]** Executing Salesforce actions after Edit/Add in Salesforce Table. The Salesforce Table windows are used to allow the form fillers to Edit and Add data, and they can also be used as a trigger, leading up to other Salesforce actions. All a user have to do is to create a new Get or Push action—Link this action to the Edit or Add window, so it will run after the record is updated or created. A user can create a form with a table and a full name field.

**[0323]** The Table will load user accounts from Salesforce when the form opens. The form filler will be required to enter a contact name in the full name field. After he updates an account via “Edit” window—this will trigger the contact’s creation in Salesforce. Follow the steps: 1—Create a new blank form. 2—Drag a Salesforce Table element. 3—Configure the columns: Account name, Phone, Website. 4—Set the Salesforce Get integration to populate the table fields. Go to ‘Properties’ panel>‘Form’ tab>‘Settings’ option>‘Salesforce Integration’ category>Get from Salesforce. Press on ‘Map fields’ button—Authenticate with Salesforce—Add object: Account and turn on the ‘Execute on form load’ checkbox. Choose ‘All’ in the ‘if multiple matches found’ radio button and enter a limit of 100. Map

the fields: Account name-->Account Name Phone-->Account Phone Website-->Website.

[0324] 5—Drag a Full Name element to the form. 6—Create a Push integration that will create a new contact in Salesforce from the values entered in the ‘full name’ field:—Authenticate with Salesforce—Choose object: Contact—Add comment: Create contact after Edit—Turn on the ‘Use in custom button’ checkbox—Action: Create—Map fields: Last name→Full Name Last Name First name→Full Name First Name—Press ‘Apply’ and save the form. 7—Select the Table element on canvas. Go to ‘Properties’ panel>‘Element’ tab>‘Settings’ option>‘Advanced’ category. Turn on the ‘Allow Edit’ checkbox and press on the ‘Configure’ button below it. 8—Make the fields editable: Account name, Phone, Website by turning on the checkboxes. 9—Press on the ‘Configure’ button of the ‘Execute Salesforce actions’ option and choose the action that will run after edit.

[0325] In this example it’s called: Contact(create)—Create contact after Edit.’ Press on the “Add” button, and the action will now be displayed in the table above. Press ‘Apply’ and save a user form. 10—Now test a user form: Publish it as URL, wait till the accounts are loaded in the Table. Fill in a full name for the test (for example: Snow White) and choose an account in the table and press on ‘Edit’ icon. Make a small change and save it. A user will now see that the Push action is running after the Update was completed. Check your Salesforce account and a user will find that an Account was updated (done from the Table Edit window) and a new contact was created (Snow White).

[0326] Salesforce Table inline editing. When a user is using the Salesforce Table in a user form a user can allow user form fillers to edit the data. A user can let them update information by pressing on the ‘edit’ icon and editing the data in the ‘Edit’ modal window and a user can also let them edit the data inline. Inline Editing means that the form filler can simply click inside a table cell and it will become editable. Following are the steps to configure Inline editing in your Table. 1—Drag a Salesforce Table on to your canvas. 2—Enter the following columns: Account Name, Account Phone, Account Website. 3—Set the Salesforce get integration to draw 100 accounts when the form loads—Authenticate with Salesforce—Select the object: Account—Turn on the ‘Execute on form load’ checkbox—Choose the ‘All’ option in the ‘If multiple matches found take ID from’ radio button And set a limit number of 100—Set the mapping: Account Name→Account Name Account Phone→Account Phone Account Website→Website—Press ‘Apply’ and save the form. 4—Select the table on canvas and go to its ‘Advanced settings’.

[0327] 5—Turn on the “Allow inline editing” checkbox. 6—Press on the ‘configure’ button and set the editable fields. 7—Turn on the ‘Editable’ checkbox in each field a user wants a user form filler to edit inline. 8—A user can also add a placeholder text for this field, that will show when the form filler clicks to edit. 9—Press ‘Apply’ and test the form. 10—Now test a user form Publish it as URL Click inside the first account’s phone field—see how it becomes editable. Change the phone number to 8888888 and it will then be updated in the Table as well as in a user Salesforce account. This feature is included in ‘Advanced tools’.

[0328] When a user form filler is updating or adding a record through the Salesforce table a user can have certain fields taken from the table field values instead of asking the

form filler to enter values in them again. A user has to: 1—Sign in to a user FormTitan account. 2—Enter a user form in the form builder. 3—Choose the relevant table level, by clicking on the level link on a canvas. 4—Go to table settings>Advanced and open the Add/Edit modal window. 5—Choose the Salesforce field a user would like to update from the dropdown at the bottom and pressing on ‘Add’. 6—Turn on the ‘Map value’ checkbox. Open the dropdown and choose the parent object and then the field a user wants to take the value from. When a user opens the mapping dropdown a user will see not only the parent objects of the table levels but also the form fields, which will allow a user to use a form field for a push.

[0329] Storing the record ID that was used in the View/Edit/Add Salesforce table modals. The Salesforce Table has 3 modal windows a user can activate and allow user form fillers to view a record, edit it or add a new record. When a user configures the window (let’s take the ‘Edit’ window, for example) a user will need to specify which of the table columns will be editable. And a user can also have the record ID of the edited item stored in a hidden field and use this after that for form logic. This can also be done in the View and Add windows. Following is an example showing how to set up the Edit window and store the record ID of the edited item. Here are the steps: 1—Create a new blank form. 2—Drag a Salesforce Table element. 3—Configure the columns: Account name, Phone, Website. 4—Now drag a hidden field below the Table Go to ‘Properties’ panel>‘Element’ tab>‘Settings’ option>‘Basic’ category and change its name to: edit\_id.

[0330] 5—Set the Salesforce Get integration to populate the table fields:—Go to ‘Properties’ panel>‘Form’ tab>‘Settings’ option>‘Salesforce Integration’ category>Get from Salesforce—Press on ‘Map fields’ button—Authenticate with Salesforce—Add object: Account—Turn on the ‘Execute on form load’ checkbox—Choose ‘All’ in the ‘if multiple matches found’ radio button and enter a limit of 100. —Map the fields: Account name-->Account Name Phone-->Account Phone Website-->Website—Press ‘Apply’ and save the form. 6—Select the Table element on canvas. Go to ‘Properties’ panel>‘Element’ tab>‘Settings’ option>‘Advanced’ category. Turn on the ‘Allow Edit’ checkbox and press on the ‘Configure’ button below it. 7—Turn on the ‘Editable’ checkboxes of user 3 column fields so the form filler will be able to update their values.

[0331] 8—Open the ‘Map record ID to HIDDEN’ dropdown and choose the hidden: edit\_id. 9—Press ‘Apply’ and save the form. 10—Now test a user form:—First make a hidden field visible for this check Go to ‘Properties’ panel>‘Form’ tab>‘Settings’ option>(View more)>‘Debug mode’ category>turn on the ‘show hidden fields’ checkbox—publish the form as URL—press on the ‘Edit’ icon to update a record and change something in one of the fields. When the update action is done a user will see that the hidden field is populated with the record ID.

Adding the Smart V to a User Form:

[0332] FIG. 70 shows for illustrative purposes only an example of adding the Smart V to a user form of one embodiment. FIG. 70 shows adding the Smart V to a user form 7000. Under elements 550 the user can select Smart V 7010. Selecting the Smart V button opens a Smart V text box 7020. The user makes an entry into the Smart V text box 7030. Other features for Smart V include a Registration link

in Smart V, Smart V customizations, Smart V logout, Smart V session configurations, Smart V supports special characters, Changing the Smart V window background color, and Date field format in Smart V of one embodiment.

[0333] Smart Validation—provides additional security for a user form. SmartV (short for Smart Validation) is a special element that adds a double verification to a user form based on data from Salesforce. It is Two Factor Authentication. When a user selects SmartV, a Login window will be created based on the fields chosen. A form filler will not be able to access the user's form. Access will only be granted after 2 steps. First step, the authentication window will appear and the form fillers will need to identify themselves, by filling in the required data. If this data they provide is found comparable with the data in the user Salesforce account, an email will be sent to them, containing a secret code.

[0334] Second step. In the second step the form fillers will need to enter the code in the authentication window. Once all data is confirmed—the form will be accessible. Smart V Session Configurations. Shared session—allows a user to set up the Smart V configuration in one form, and then share this configuration with other forms. Allowed Session—allows a user to set a time frame (in minutes) for your Smart V session. Captions—allows a user to customize the text in the verification window. Email Settings—allows a user to customize the email containing the secret code.

[0335] Adding the Smart V to a user form includes the following steps: 1—Enter a form in the form builder. 2—Drag a Smart V element from “Elements panel”>“Salesforce”>“SmartV”. 3—Go to “Properties” panel>“Element” tab>“Settings” option>“Basic” category. 4—Press on the “Map fields” button to configure a validation. 5—Since the validation is done based on the data in your Salesforce account a user will first need to authenticate with Salesforce. 6—Once authenticated press on the “Configure Salesforce integration” button. 7—Choose the Salesforce object to use for this validation—in this example: Contact. 8—Select the fields for the validation by choosing them from the “Add field for display” dropdown, and then press on “Add” button. A user must add an “email” field so an email with a secret code could be sent to the form filler later on, in the second step of the validation. In this example the following fields may be selected from the Salesforce contact object: Email, First name, Secret, Contact ID, Last name.

[0336] 9—These fields will now be displayed in a table that contains 3 columns: Email, Value and Secret. The user will now have to choose:—which of the fields will be used for drawing the email address (“email”)—which of the fields will be used to hold returned value (for example “contact id”)—which of the fields will be used to store the secret code (for example “secret”). 10—Once this is done press on “Apply” and save the form. 11—In order to do a live check the user will need to create a test contact in the Salesforce account, beforehand and make sure it contains:—the form owner user first and last name, —a “Secret” field (create a custom text field for this)—the form owner user email address (so that the code is sent to the form owner user). The form owner user will now see the Smart validation window.

[0337] Choose a real contact from the Salesforce account and enter its details in this window: the email address, first name and last name. 12—Publish the form as “URL” to test it. The user will now see the Smart Validation window. Enter the email, last name and first name of the Test Contact. A

second window will appear to enter a code. 13—Go to the user email inbox and copy the code that was sent to the user. 14—Paste this code in the window and submit—and the form will open right after. Smart V logout.

[0338] The Smart V is a 2 factor authentication a user can add to a user form, and it allows the user to make sure that only those contacts the user has selected will be able to access the user form. A user can also specify how long the form filler's session will last before the form filler will need to authenticate again. A form filler can elect to Logout from the session. The form owner user can drag and drop a button element and changing its type to “SmartV Logout” to allow a form filler to sign out of the session of one embodiment.

Push the Data to Salesforce Using a Custom Button:

[0339] FIG. 71 shows for illustrative purposes only an example of push the data to Salesforce using a custom button of one embodiment. FIG. 71 shows a feature to push the data to Salesforce using a custom button 7100. Under elements 550 the user can select button 850 on the menu. The button selection opens an input box 7130. For example the button selection can create a submit button 7140 for a form of one embodiment.

[0340] A Salesforce Action button can be used for Push functions. Push the data to Salesforce using a custom button Custom Salesforce buttons are buttons designed especially for working with Salesforce. They are located in the “Elements panel” under “Salesforce” category and their purpose is to allow users to push the data to Salesforce without having to submit the form. Following is an example of how to add two custom buttons to the form: one to push data to the “Account” object, and the other to push data to the “Contact” object. Follow the steps: 1—Create a new blank form in the form builder 2—Drag a textbox. Change its label to “Account Name”. Drag a full name element and change its label to “Contact full name”. 3—Go to “Properties” panel>“Form” tab>“Settings”>Salesforce integration”>Push—authenticate with Salesforce 4—A user will now need to configure 2 push lines: the first for account and the second for contact. Account—In the “Salesforce object settings” category—choose the “Account” object—Turn on the “Use in custom button” checkbox—this will make this push triggered by a custom button—In the “Action” category—The default action is “create”—leave it as is—map the fields—in this case only one (account name).

[0341] 5—Contact—Now press on “Add object” button to add the contact. —In the “Salesforce object settings” category—choose the “Contact” object—Turn on the “Use in custom button” checkbox—Map the contact fields and save it all. A user will now see 2 lines of push. 6—Add the custom buttons to a user form—Drag a button from “Elements” panel>“Salesforce” category>“Button” Place it next to the account textbox—a window will open and a user will need to connect this button to the right action: Open the “Process push” dropdown, choose “Account (create) and press on “Add” button next to it—Press on “Apply”—Select the button on canvas and change the text on it to “Push account”. (in “Properties” panel>“Element”>“Settings”>“General”)

[0342] Please note—if a user is working on a multi-page form then another field will be added to this window: ‘Go to Page’ dropdown. A user can using this field add another functionality to this Action button, so it doesn't just push data to Salesforce, but it also moves the form filler to a different page after that. Read how to do this 7—Drag

another button from ““Elements” panel>“Salesforce” category>“Button” Place it next to the account textbox—configure this button in the same way a user did with the previous button. A window will open and a user will need to connect this button to the right action open the “Process push” dropdown, choose “Account (create) and press on “Add” button next to it. Press on “Apply” and select the button on the canvas and change the text on it to “Push contact”.

In “Properties” panel>“Element”>“Settings”>“General”. 8—Save the form now publish the form as URL to check it. Enter an account name and press on the custom Salesforce button next to it—Enter the user Salesforce account and check that this data was added in the Account object. Enter a contact name and press on the custom Salesforce button next to it—Enter the user Salesforce account and check that this data was added in the contact object.

**[0343]** Redirect in Custom Push/Get With Action Button. When a user uses the Salesforce Action button to create a custom Push or Get a user doesn’t have to submit the form because the action button acts as trigger. If a user wants to have a user form redirect to another URL after this custom Get or Push—a user can add this functionality in the Action button configuration. Following is an example to show a user how to redirect a user form after a custom Get. In this example the form filler will enter an account name and press on the custom button, and based on the account the form filler entered the phone number will be drawn and a URL will be stored in the form’s hidden field—all this populated from the Salesforce ‘Account’ object.

**[0344]** After this, the form will redirect to a different URL (in this case another FormTitan published form) based on the URL in the hidden field and params will also be added at the end of the URL to pre-fill some fields in it. Before starting—create test data. In order to follow this example a user will need to create some test data: A—In a user form Titan account—Create a new blank form—call it “test redirect”—The main form will redirect to the new blank form then drag a textbox in it. Change its label text to “Account” and check what its Field ID is. (When it is selected on a canvas a user will see its ID inside the “Settings” tab)—Press on the “Publish” icon in the top toolbar and copy the form URL to redirect to it. B— In your Salesforce account—set up a test record in the “Account” object—create a new account, Give it a name, a phone number and insert the URL from the previous bullet in the Account Description field.

**[0345]** Create a user main form with a custom Get 1—Create a new blank form. 2—In the form builder—Give a user form a name: custom get (“Properties” panel>Form>Settings>General>Basic>General). 3—Drag the following elements on to the canvas:—Textbox—for account field (From “Element” panel>“Basic” category>“Input” sub category>Textbox)—Textbox—for phone field (From “Element” panel>“Basic” category>“Input” sub category>Textbox)—hidden field—to store a URL (From “Element” panel>“Widgets” category>Hidden). 4—Select the first textbox and change its label text to: Account Select the second textbox and change its label text to: Phone Select the hidden field and change its name to: id\_URL. 5—Delete the default submit button, which is already on canvas. 6—Create a Salesforce Get integration:—Go to “Properties” panel>“Form” tab>“Salesforce Integration”>Get from Salesforce—Push on the “Map fields” button—Authenticate with Salesforce—

Choose an object: Account and turn on the “Use in custom button” checkbox—Set a condition: Account name equals account—Map the fields: Phone-->Account Phone id\_URL-->Account Description.

**[0346]** 7—Drag a Salesforce Action button (From “Element” panel>“Salesforce” category>Button) Once on canvas, the configuration window will open and a user will need to map the custom button to Get action—Open the “process push” dropdown and chose the “Account (Get)” option.—Press on the “add” button and the action will be added in the grid at the top.—Turn on the “Redirect after finish” checkbox—this will open a few fields a user will need to set:—URL: Choose the form field a user would like to draw the URL from (in this case it will take the URL from the hidden field, which is mapped to the account description field in Salesforce).—Open in: a user will be able choose between Self and Parent.—Set parameters—a user can choose parameters that will be added to the URL in order to pre-fill fields in the form. In this case it will turn on the Press on the “Set parameters” button and Turn on the “Use” checkbox in the rows of the fields a user would like to prefill. Enter a Param name—in this case it will use the ID of the field a user wants to pre-fill in the form a user redirected to: fld2. 8—Select the button and change the text on the button to: Get Now.

**[0347]** 9—Now test a user form:—Publish it as URL and test it—Enter the name of the test account in Salesforce—a user will now see the phone number populated based on it—the form will redirect automatically to the test form a user created and a user will see that the first textbox was prefilled.—If a user will look closely at the URL and a user will see it is the same one the user inserted in the user Salesforce account, and the params are added at the end of it.

**[0348]** Custom Salesforce buttons in a Multi-page Form. Custom Salesforce button pushes data and moves the form filler to a different page. A user can use the Action button in a multi-page form. In the multi-page form this button can not only push data to Salesforce but it can also move the form filler to the next/previous page.

**[0349]** Following is an example of the steps. 1—Create a new blank form in the form builder. 2—Drag a textbox. Change its label to “Account Name”. 3—Go to “Properties” panel>“Form” tab>“Settings”>Salesforce integration”>Push—authenticate with Salesforce. 4—A user will now need to configure the push integration—In the “Salesforce object settings” category—choose the “Account” object—Turn on the “Use in custom button” checkbox—this will make this push triggered by a custom button (an Action button)—In the “Action” category—The default action is “create”—leave it as is—map the fields—in this case only one (account name)—A user will now see one integration line was created—Press on “Apply”. 5—Save the form. 6—Add the custom buttons to a user form—Drag a button from ““Elements” panel>“Salesforce” category>“Button” Place it next to the account textbox—a window will open and a user will need to connect this button to the right action: Open the “Process push” dropdown, choose “Account (create) and press on “Add” button next to it—Press on “Apply”—Select the button on canvas and change the text on it to “Push account”. (in “Properties” panel>“Element”>“Settings”>“General”)

**[0350]** When a user is working on a multi-page form then another field will be added to this window: ‘Go to Page’

dropdown. Using this field a user can add another functionality to this Action button, so it doesn't just push data to Salesforce, but it also moves the form filler to a different page after that. —Select the option "Next Page" in the dropdown. 7—Save the form now publish the form as URL to check it. Enter an account name and press on the custom Salesforce button next to it—See that a user are forwarded to the next page in the form. Also, enter the user Salesforce account and check that this data was added in the Account object.

[0351] Triggering Salesforce actions from a page break. The page break trigger is a new option to run Salesforce actions, and it works in multi-page forms. A user could have "Get" and "Push" actions either triggered on load (when the form opened), on submit (when the form was sent) or when a custom Salesforce Action button was pressed. The multi-page form makes the page break buttons act as triggers. It can help make the overall flow of the form faster. What's special about the multi-page form is that loading all the pages data in one go is not a necessity because the user can only view one page at a time anyway, so if a user break down Salesforce actions per page a user can cut loading time short. In order to use this option, all a user has to do is:—Select the page break element in a specific page, and attach the Salesforce actions to this page by setting integration lines as—and making them work via custom button (which in this case means the page break buttons). Define when the trigger will work:—If it will work only the first time a user reaches the page (this is the default trigger)—If it will work only if a user arrives at the page from a Next button—If it will work only if a user arrives at the page from a Back button—If it will work every time a user reaches the page. When a user adds a page break to a user form the default trigger state in all the form pages is "Load first time". But this can be changed to one of the other options listed above.

#### Set Salesforce Action for Custom Button:

[0352] FIG. 72 shows for illustrative purposes only an example of set Salesforce action for custom button of one embodiment. FIG. 72 shows the user can set Salesforce action for custom button 7200. A set Salesforce action for button web page shows a Salesforce action 7220 a user can select including account (create) 7225, remove 7210, return value to 7230, select an object 4030, process push 7250 and select an object 7255 of one embodiment.

[0353] A Salesforce action button can easily be added to a user form to get/push data right then and there, without having to submit the form. However, when setting the button to act on a repeated section it applies to all of the data in all of the repeated lines. If a user wants to have a user form filler press on the custom button in a specific repeated line and have it apply to this line only—a user can easily do so. Follow these steps: 1—Create a new blank form. 2—Drag a section element and make it wider. 3—Drag a textbox, a numeric and a button into the section. 4—Change the textbox label to: account Go to 'Properties' panel>'Element' tab>'Settings' option>'Basic' category>Label.

[0354] 5—Change the numeric label to num. 6—Change the button text to: push this item now. 7—Select the section and make it Repeated. Go to 'Properties' panel>'Element' tab>'Settings' option>'Basic' category>Repeated. 8—Now configure the Salesforce push integration 'Properties' panel>'Form' tab>'Settings' option>'Salesforce integration' category>'Push to Salesforce'—Press on the 'Set notifica-

tion' button—Authenticate with Salesforce—Choose the account object: Account—Turn on the 'Use in custom button' checkbox—in the 'Action' category—leave the 'Create' option selected—Map the fields: Account Name-->account Num>num—Press 'apply', then 'Finish' and save the form. 9—Select the 'button' and change its type to 'Salesforce Action'.

[0355] 10—Press on the 'Set' button to have the integration line work with your custom button. 11—Open the 'process push or get' dropdown and select the Salesforce action. 12—Press on 'Add' and it will be added to a table above. 13—Turn on the 'Execute per repeated item' checkbox. 14—Press 'Apply' and save a user form. 15—Now it's time to test a user form:—First select Section and change visible items to 2 and turn on the 'Populate items' checkbox—Publish it as URL—Enter the first item line in the section: an account name and number: test 1, 1 and test 2, 2.—Press on "Add" and enter another item line with data: test 3, 3. Choose a specific line (for example second line) and press on the 'push' button in that line. Check a user Salesforce account and a user will see that only that line of data was added as a new record.

#### Additional Features:

[0356] Voice recognition can be added to the bi-directional voice command and interactive form building. Recognizing the user's voice when employing the voice element of the form builder feature of the customized customer relationship management platform provides an additional level of security. If a voice analyzer of the customized customer relationship management platform does not match the current voice being received to the voice analysis of the authorized user then it closes the form builder and proceeds to another security check.

[0357] Iris recognition can be added to the customized customer relationship management platform security features. The user for example can use the camera of a smart phone for capturing their iris image. The captured iris image is processed against the authorized user iris image stored by the customized customer relationship management platform security features. A positive match processes the request to the form builder, if not no access is provided.

[0358] The iris recognition must be passed positively to allow a user to use their eye movements to navigate the form using eye commands to create the form and/or fill in the form. The eye commands can move for example a textbox element on the form to a user desired position. The eye command can be used to select a form input element and the voice command to make the input audibly when filling out the form.

[0359] Human gestures, like hand movements can be tracked using a proximity sensor, accelerometer and gyroscope to for example change form pages, zoom in and out, change the form builder feature for a next operation and other user desired operations.

[0360] Computer vision and video recognition of facial expressions can be captured for example using a smart phone camera. The captured facial expressions can be incorporated in the sentiment analyze feature and used separately for navigating a form or features selection. A raise eyebrow can for example be given an operation selection by the user which is stored and can then be used by the user to for example visually focus on the submit button then raise an eyebrow to activate a click to submit the form.

[0361] A user who has created a number of forms using the form builder and activate a machine learning feature. The machine learning feature can analyze for each type of form the common elements selected and placement position in the form. The user can then automate an initial form creation to include in the chosen position those common elements. This saves time for the user and reduces the duplication of effort to arrive at the same or similar form layout.

[0362] The customized customer relationship management platform application can configure for example a smart phone accelerometer for motion sensing the phone position for example portrait or landscape to activate the mini mode display sizing conversion to fit the phone positioning. The accelerometer may also be used for adjusting augmented reality apps for example in a help situation where a user is chatting with a FormTitan associate and an augmented reality app is allowing the associate to follow the user's selections, positioning of an element and other movements in real time directly to better assist the user in understanding where they may have been incorrectly using the form builder feature. The accelerometer is also used to measure a user driving speed to estimate an ETA to a meeting site as described in FIG. 66.

[0363] The accelerometer may also be used activate features. For example the user can select a smart phone physical movement for example a quick side to side shaking movement to activate a publish form feature. The user for example could select a top to bottom shaking movement to activate a form submit button.

[0364] A magnetometer for example in a smart phone senses where a phone is in physical space. It is used in mapping a user's location and travel. The magnetometer is used with an accelerometer and GPS unit to determine where a user is located.

[0365] A smart phone digital barometer coupled with a meteorological app can alert a user if in their travels they may encounter inclement weather changes. This could cause delays in the meeting scheduled. Gather barometric data on a remote user's location may also explain a remote users difficulties in a clear connection using the customized customer relationship management platform application as there may be for example lighting storms in the area that are disrupting internet, cellular or WI-FI communications.

[0366] Biometrics can be integrated into the customized customer relationship management platform application to using sensors to provide levels of enhanced security by capturing and validating human related metrics including Finger Print recognition, IRIS (eye) scanning and full facial recognition. Additionally, biometric sensors can be used to collect a user's heart rate and SpO<sub>2</sub> (the estimate of arterial oxygen saturation) for use within a vendors 'health' application form and in the sentiment analysis.

[0367] Cognitive services can be integrated with the customized customer relationship management platform and application. Cognitive services can include for example Vision-Image-processing algorithms to smartly identify, caption and moderate user and other pictures; Speech-Convert spoken audio into text, use voice for verification, or add speaker recognition to a user's app; Knowledge-Map complex information and data in order to solve tasks such as intelligent recommendations and semantic search; Search-Add Bing Search APIs to user apps and harness the ability to comb billions of webpages, images, videos, and news with a single API call; and Language-Allow user apps to

process natural language with pre-built scripts, evaluate sentiment and learn how to recognize what users wants.

[0368] Because the Cognitive Services APIs harness the power of machine learning, and brings advanced intelligence into the FormTitan products without the need to have a team of data scientists on hand. Enhance security with a face. Use the Face API to verify a selfie for smart authentication. Signing in with visual identity verification is becoming an additional security layer for many industries. Microsoft's Face API can compare portraits giving it amazing flexibility in uncontrolled scenarios. Express dramatic moments in an instant. Quickly retrieve surprised, happy, or sad celebrity images out of millions by combining multiple APIs. Search through video frames to pull out the perfect moment for user content. Never let an abundance of untagged footage be a deterrent for a user ability to serve up contextual content when the user needs it. Engage customers through chat. Bring together cognitive service APIs and Bot Framework to engage a user audience on a whole new level. Build a bot that embodies a user brand, addresses a user's customers' main questions and escalates to a human operator if needed.

[0369] Allow great voice interactions with speech customization. With the custom speech service a user can build great voice interactions between user systems and users. Enhance speech recognition using background noise reduction and complex technical word training. Emotion preview with the Emotion API takes a facial expression in an image as an input, and returns the confidence across a set of emotions for each face in the image, as well as bounding box for the face, using the Face API. Computer vision—this feature returns information about visual content found in an image. Use tagging, descriptions, and domain-specific models to identify content and label it with confidence.

[0370] Language understanding preview-language understanding provides simple tools that allow a user to build a user's own language models, which allow any application or bot to understand user commands and act accordingly. Speaker recognition-Identify who is speaking. Input audio of the unknown speaker is paired against a group of selected speakers, and in the case there is a match found, the speaker's identity is returned. Cognitive services employ Artificial Intelligence (AI) can include integrating 3rd party applications including for example Microsoft AI Platform Services that compose intelligent applications, customized to a user's organization's availability, security, and compliance requirements. Infrastructure Services and tools backed by best-of-breed infrastructure enterprise grade security, availability, compliance and manageability. Tools can include for example leveraging a set of comprehensive tools and frameworks to build, deploy and operationalize AI products and services at scale.

[0371] Integrating 3<sup>rd</sup> party applications can include cloud computing including Azure cloud platform, Azure securityRely on a trusted cloud security foundation, Azure global infrastructureAchieve global reach and support local compliance, Featured Explore, Virtual MachinesProvision Windows and Linux virtual machines in seconds, Windows Virtual DesktopDeliver a virtual desktop experience to any device at cloud scale, Azure SQL DatabaseManaged relational SQL Database as a service, App ServiceQuickly create powerful cloud apps for web and mobile, Azure Cosmos DBGlobally distributed, multi-model database for any scale, Machine LearningOpen and elastic AI develop-

ment spanning the cloud and the edge, Azure Kubernetes Service (AKS)Simplify the deployment, management, and operations of Kubernetes, FunctionsProcess events with serverless code, Cognitive ServicesAdd smart API capabilities to allow contextual interactions, and Blockchain WorkbenchConnect a user's blockchain to the cloud without the heavy lifting.

[0372] Integrating 3<sup>rd</sup> party applications can include applications using artificial intelligence capabilities including Cognitive ServicesAdd smart API capabilities to allow contextual interactions, Azure Bot ServiceIntelligent, serverless bot service that scales on demand, Azure DatabricksFast, easy, and collaborative Apache Spark-based analytics platform, Machine LearningOpen and elastic AI development spanning the cloud and the edge, Cognitive Services—Search APIsHarness the ability to comb billions of webpages, images, videos, and news with a single API call, Cognitive Services—Language APIsProcess natural language with pre-built scripts, evaluate sentiment, and learn to recognize intent, Cognitive Services—Vision APIsUse Image-processing algorithms to smartly identify, caption and moderate user pictures, Cognitive Services—Speech APIsConvert speech to text or text to speech, translate text or audio, or add speaker recognition to a user app, and Cognitive Services—Knowledge APIsMap information and data in order to solve complex tasks.

[0373] Integrating 3<sup>rd</sup> party applications can include Analytics Analytics Gather, store, process, analyze, and visualize data of any variety, volume, or velocity, SQL Data WarehouseElastic data warehouse as a service with enterprise-class features, Azure DatabricksFast, easy, and collaborative Apache Spark-based analytics platform, HDInsight-Provision cloud Hadoop, Spark, R Server, HBase, and Storm clusters, Data FactoryHybrid data integration at enterprise scale, made easy, Machine Learning Open and elastic AI development spanning the cloud and the edge, Stream Analytics Real-time data stream processing from millions of IoT devices, Data Lake Analytics Distributed analytics service that makes big data easy, Azure Analysis ServicesEnterprise-grade analytics engine as a service, and Event Hubs—Receive telemetry from millions of devices.

[0374] Integrating 3<sup>rd</sup> party applications can include Compute Compute Access cloud compute capacity and scale on demand—and only pay for the resources users use, Virtual MachinesProvision Windows and Linux virtual machines in seconds, Virtual Machine Scale SetsManage and scale up to thousands of Linux and Windows virtual machines, Azure Kubernetes Service (AKS)Simplify the deployment, management, and operations of Kubernetes, FunctionsProcess events with serverless code, Service FabricDevelop microservices and orchestrate containers on Windows or Linux, App ServiceQuickly create powerful cloud apps for web and mobile, Container InstancesEasily run containers on Azure without managing servers, BatchCloud-scale job scheduling and compute management, and Azure Batch AI Easily experiment and train user deep learning and AI models in parallel at scale.

[0375] Integrating 3<sup>rd</sup> party applications can include Containers Containers Develop and manage user containerized applications faster with integrated tools, Azure Kubernetes Service (AKS)Simplify the deployment, management, and operations of Kubernetes, Container InstancesEasily run containers on Azure without managing servers, Service FabricDevelop microservices and orchestrate containers on

Windows or Linux, Container RegistryStore and manage container images across all types of Azure deployments, App ServiceQuickly create powerful cloud apps for web and mobile, Web App for ContainersEasily deploy and run containerized web apps that scale with a user business, and BatchCloud-scale job scheduling and compute management.

[0376] Integrating 3<sup>rd</sup> party applications can include Databases Databases Support rapid growth and innovate faster with secure, enterprise-grade, and fully managed database services, Azure Cosmos DBGlobally distributed, multi-model database for any scale, Azure SQL DatabaseManaged relational SQL Database as a service, Azure Database for MySQLManaged MySQL database service for app developers, Azure Database for PostgreSQLManaged PostgreSQL database service for app developers, Azure Database for MariaDBManaged MariaDB database service for app developers, SQL Server on Virtual MachinesHost enterprise SQL Server apps in the cloud, SQL Data WarehouseElastic data warehouse as a service with enterprise-class features, Azure Database Migration ServiceSimplify on-premises database migration to the cloud, and Redis CachePower applications with high-throughput, low-latency data access.

[0377] Integrating 3<sup>rd</sup> party applications can include Developer Tools Developer Tools Build, manage, and continuously deliver cloud applications—using any platform or language, Visual StudioThe powerful and flexible environment for developing applications in the cloud, Visual Studio CodeA powerful, lightweight code editor for cloud development, SDKsGet the SDKs and command-line tools a user needs, Azure DevOpsServices for teams to share code, track work, and ship software, CLIsBuild, deploy, diagnose, and manage multi-platform, scalable apps and services, Azure PipelinesContinuously build, test, and deploy to any platform and cloud, Azure Lab ServicesSet up labs for classrooms, trials, development and testing, and other scenarios, Azure DevTest LabsQuickly create environments using reusable templates and artifacts, and Developer tool integrationsUse the development tools a user can know—including Eclipse, IntelliJ, and Maven—with Azure.

[0378] Integrating 3<sup>rd</sup> party applications can include DevOps DevOps Deliver innovation faster with simple, reliable tools for continuous delivery, Azure DevOpsServices for teams to share code, track work, and ship software, Azure PipelinesContinuously build, test, and deploy to any platform and cloud, Azure BoardsPlan, track, and discuss work across user teams, Azure ReposGet unlimited, cloud-hosted private Git repos for a user project, Azure ArtifactsCreate, host, and share packages with a user team, Azure Test PlansTest and ship with confidence with a manual and exploratory testing toolkit, Azure DevTest Labs—Quickly create environments using reusable templates and artifacts, and DevOps tool integrationsUse user favorite DevOps tools with Azure.

[0379] Integrating 3<sup>rd</sup> party applications can include Identity Identity Manage user identities and access to protect against advanced threats across devices, data, apps, and infrastructure, Azure Active DirectorySynchronize on-premises directories and allow single sign-on, Azure Active Directory B2CConsumer identity and access management in the cloud, Azure Active Directory Domain ServicesJoin Azure virtual machines to a domain without domain controllers, and Azure Information ProtectionBetter protect user sensitive information—anytime, anywhere.

[0380] Integrating 3<sup>rd</sup> party applications can include Integration Integration Seamlessly integrate on-premises and cloud-based applications, data, and processes across a user enterprise, Logic AppsAutomate the access and use of data across clouds without writing code, Service BusConnect across private and public cloud environments, API ManagementPublish APIs to developers, partners, and employees securely and at scale, Event GridGet reliable event delivery at massive scale.

[0381] Integrating 3<sup>rd</sup> party applications can include Internet of Things Internet of Things Bring IoT to any device and any platform, without changing a user infrastructure, IoT HubConnect, monitor and manage billions of IoT assets, IoT EdgeExtend cloud intelligence and analytics to edge devices, IoT CentralExperience the simplicity of SaaS for IoT, with no cloud expertise required, IoT solution acceleratorsCreate fully customizable solutions with templates for common IoT scenarios, Azure SphereSecurely connect MCU-powered devices from the silicon to the cloud, Time Series InsightsExplore and analyze time-series data from IoT devices, Azure MapsSimple and secure location APIs provide geospatial context to data, FunctionsProcess events with serverless code, and Event GridGet reliable event delivery at massive scale.

[0382] Integrating 3<sup>rd</sup> party applications can include Management Management and Governance Simplify, automate, and optimize the management and compliance of user cloud resources, Microsoft Azure portalBuild, manage, and monitor all Azure products in a single, unified console, Cloud ShellStreamline Azure administration with a browser-based shell, Azure AdvisorYour personalized Azure best practices recommendation engine, Azure BackupSimple and reliable server backup to the cloud, Cost ManagementOptimize what a user spends on the cloud, while maximizing cloud potential, Azure PolicyImplement corporate governance and standards at scale for Azure resources, Azure MonitorHighly granular and real-time monitoring data for any Azure resource, Azure Site RecoveryOrchestrate protection and recovery of private clouds, and SchedulerRun user jobs on simple or complex recurring schedules.

[0383] Integrating 3<sup>rd</sup> party applications can include Media Media Deliver high-quality video content anywhere, any time, and on any device, Media ServicesEncode, store, and stream video and audio at scale, EncodingStudio grade encoding at cloud scale, Azure Media PlayerA single layer for all user playback needs, Live and On-Demand StreamingDeliver content to virtually all devices with scale to meet business needs, Media AnalyticsUncover insights from video files with speech and vision services, Content ProtectionSecurely deliver content using AES, PlayReady, Widevine, and Fairplay, Video IndexerUnlock video insights.

[0384] Integrating 3<sup>rd</sup> party applications can include Microsoft Azure Stack Microsoft Azure Stack. Microsoft Azure Stack is an extension of Azure—bringing the agility and innovation of cloud computing to user on-premises environment and enabling the only hybrid cloud that allows a user to build and deploy hybrid applications anywhere.

[0385] Integrating 3<sup>rd</sup> party applications can include Migration Migration Simplify and accelerate user migration to the cloud, Azure MigrateEasily discover, assess, right-size, and migrate user on-premises VMs to Azure, Azure Site RecoveryOrchestrate protection and recovery of private clouds, Azure Database Migration ServiceSimplify

on-premises database migration to the cloud, Data BoxSecure, ruggedized appliance for Azure data transfer, and Cost ManagementOptimize what a user spends on the cloud, while maximizing cloud potential.

[0386] Integrating 3<sup>rd</sup> party applications can include Mobile Mobile Build and deploy cross-platform and native apps for any mobile device, Mobile AppsBuild and host the backend for any mobile app, Notification HubsSend push notifications to any platform from any back end, Visual Studio App CenterShip apps faster by automating application lifecycles, XamarinCreate cloud-powered mobile apps faster, Azure MapsSimple and secure location APIs provide geospatial context to data, and API AppsEasily build and consume Cloud APIs.

[0387] Integrating 3<sup>rd</sup> party applications can include Networking Networking Connect cloud and on-premises infrastructure and services to provide user customers and users the best possible experience, Virtual NetworkProvision private networks, optionally connect to on-premises datacenters, Load BalancerDeliver high availability and network performance to user applications, Application GatewayBuild secure, scalable, and highly available web front ends, VPN GatewayEstablish secure, cross-premises connectivity, DNS Hosting of a user DNS domain, Content Delivery NetworkEnsure secure, reliable content delivery with broad global reach, Azure DDoS ProtectionProtect user applications from Distributed Denial of Service (DDoS) attacks, Traffic ManagerRoute incoming traffic for high performance and availability, and Azure Front Door ServiceScalable, security-enhanced delivery point for global, microservice-based web applications.

[0388] Integrating 3<sup>rd</sup> party applications can include enterprise protection from advanced threats across hybrid cloud workloads, security management and allow advanced threat protection across hybrid cloud workloads, maintain control of keys and other secrets, secure, scalable, and highly available web front ends, protect user sensitive information—anytime, anywhere, secure, cross-premises connectivity, on-premises directories and allow single sign-on, protection for applications from distributed denial of service (DDoS) attacks, and advanced threat protection, detection and investigation of advanced attacks on-premises and in the cloud.

[0389] Integrating 3<sup>rd</sup> party applications can include massively scalable cloud storage application platforms for user data, apps, and workloads, object storage for unstructured data, rarely accessed data, storage for apps according to traffic, storage for standard SMB 3.0 protocol, secured disk options supporting virtual machines, scalable data lake storage, data transfer, and other storage resources.

[0390] Integrating 3<sup>rd</sup> party applications can include application to deploy, and scale powerful web applications quickly and efficiently, create and deploy mission critical web apps at scale, publish APIs to developers, partners, and employees securely and at scale, secure content delivery, reliable content delivery with broad global reach, managed search-as-a-service, real-time web functionalities and other application. Integrating 3<sup>rd</sup> party applications can include cognitive services with apps, websites and bots with intelligent algorithms to see, hear, speak, understand and interpret a user needs through natural methods of communication.

[0391] The customized customer relationship management platform method and devices includes other features,

elements and form building features including Add company info to billing, Add script after button completes its task, Address populating an online map—Additions, Address validation for New Zealand and Australia, Advanced Field Conditional Logic, Allow alignment in the Calculated field, Allow Calculator to use First or Last name of fullname, Area selection, Auto focus in Lookup search box, Auto position for section with tabs, Billing—update billing, Calendar supports Russian and Hebrew, Categories added to My Forms, category in form settings: Head, category in get: Meta caching, changing instance Confirmation message, Condition and Value rule based on word count, Condition based on advanced values, Conditional mapping in Repeated and Files, Convert lead to contact, Currency format in payment integration, Custom value rule messages, Cut Action, Date and DateTime Additions, Debug Mode for checking hidden fields, Disallow items of Picklists in Dropdown, Radio button, Checkbox/Multiple dropdown, Display full error details for users, Document generation (single/bulk) directly from Salesforce, Draft button—Hide confirmation, Dropdown—Advanced navigation mode, Dropdown default selection, Dropdown for dynamic navigation in multi-page forms, DST Support, Edit Hidden field value in my submission, Effected by, Effected by option Extended, Element Type Displayed in Basic Settings, Element: Slider, Elements Search, Else added to the Conditional mapping, Empty options in Dependency Picklist, Allowed and Disabled for Lookup Condition, Enter Key moves to next field like tab, Enter key working in Duplicate form, Entry duplication, Excess buttons removed from repeated section, Executing actions after Delete in Salesforce Table, Executing actions after View/Edit/Add in Salesforce Table, Export added to Salesforce Table, Export more than 1000 submissions, Field Condition/Value re-design and enhancement, Field tag, File upload additions, File upload compression, File upload new properties, File upload size limit in Value rule, File upload size limit to 30 MB, filter to Editors, filter to Field Dropdowns, Find action supports Page Break, Form Builder layout made lighter, form settings category: Debug Mode, Form settings: fields affecting and fields effected by, Form URL starting with a number, Function mode to String Calculated Field, Geo localization for Address and online maps, Go to page Button based on Hidden, online map based on input of address, Hebrew country list in Address field, Hidden field new system value: browser, Hidden Fields Can Be Resized, Hidden fields in auto-layout for tablet and mobile, Hidden fields in Custom editors, hover/selected effects for section, HTML mode for emails, Ignore Mandatory for Debug, Image capture orientation, Image slider auto play with interval in form settings, Inline editing in the Salesforce Table, language to custom translation: Chinese, likert type: dropdown, Limit number of rows in a Salesforce table based on parameters, limit:ip whitelist, Link to Draft form in my submissions, Linkable columns in Lookup, links in Dashboard/My-forms, Lookup—add placeholder, Mapping only reference fields in your Table, Multi Dropdown placeholder, Multi Picklist Include/Exclude with OR operator, My account—new tab: mail, My submissions—export limits, My submissions automatically refreshed after action from Sub menu, Open Redirect in modal window, Optimized CSS compilation, option for spacing: Make same space, option in Conditional logic: Force execution, option to Hide the error summary modal window, options for Bulk Apply in Condition, options to Field Condition Results,

Page break Triggers in multi-page forms, Page break with flexible next/prev buttons, Parameters added to Redirect URL option in Form Condition, Pass submission id in hyperlink with parameter, Populate Date Time fields with Parameters from URL, Populate items in Section (for Number of visible items), Populate Lookup field from URL, Predefined lists including all languages—Hebrew, all languages, world countries, USA states, marital status, and months of the year (long), Progress indicator—Gradient, Property for Hidden: Use system value, Property in Account Settings: Compressed mode, property in Section: Repeat margin, Reading field values from an external script, Resized modal windows in my submissions, Save and resume—returns to the same page, Save and resume in Hebrew, Search with Lookup, Section overflow—scroll, Section Repeat with mapped data, Set order of fields in sf table view window, Set Time in Time and Date Time fields, Set Value From, Set values in fields from scripts, Setting a default year for date/datetime fields, Setting current date/time as default values in Date/Time fields, Show Hidden Fields in Debug Mode showing all fields, Show item number in repeated section, Silent mode property added to the Inline Edit window, Star rating supported in Calculated field, Submission ID in mapping of nested child, Support for Dependency Picklist, Supporting Slider Disable/Allow/Read only, system value in Hidden: IP, tab added to My account: Company info, “thank you” message box title text, Tooltip for disabled text fields, Tooltip style properties including Tooltip Label color, Tooltip Background color, Font size and Font family, Tooltip supports CSS, Upgrading our servers, Uploading/Downloading modal added to submissions page, Use mask value, User Requests, Using parameters to reach a specific page in multi-page form, Using the Short State name in Address mapping, Value rule—Min count of words, Value Rule: Address Validation, Versatile Section, When form is updated show asterisk in browser tab, and work mode: Mini of one embodiment.

#### HL7 EHR Cross Platform Application:

[0392] FIG. 73 shows for illustrative purposes only an example of an HL7 EHR cross platform application of one embodiment. FIG. 73 shows the cross platform integration 140 an application for HL7 electronic health records (EHR) using custom FormTitan created forms from three different medical office. A MD office #1 7310 FormTitan cross HL7 platform integrated EHR #1 form 7311 includes patient data for example patient 7312, DOB 7313 and allergies 7314. A MD office #2 7320 FormTitan cross HL7 platform integrated EHR #2 form 7321 includes the same patient data however listed under different title and in a different sequence for example last name 7322, first name 7323, birth date 7324 and pharma reactions 7325.

[0393] A current MD EHR query 7330 for the electronic health records for the same patient data is made by MD office #3 7331. MD office #3 7331 has created a FormTitan cross HL7 platform integrated EHR #3 form 7332. The FormTitan cross HL7 platform integrated EHR #3 form 7332 from the MD office #3 7331 includes the same patient data however listed under different title and in a different sequence for example full name 7333, DOB 7334, and med allergies 7335. The current MD EHR query 7330 would normally not be able to automatically transfer the patient data to a non-FormTitan form as the listing titles would not

match. The MD office #3 staff would have to download the other forms and manually enter the patient data from the non-FormTitan forms.

[0394] In this exampled instance since all of the forms were created using FormTitan the current MD EHR query **7330** routed through the platform network attached storage cloud **320** can use the customized customer relationship management platform network **300** network server to access the electronic health records (EHR) **7300**. The customized customer relationship management platform network **300** network server can perform a search for the existing EHR FormTitan forms records. The network server using the FormTitan pdf mapping elements **7340** used when the two existing FormTitan cross HL7 platform integrated EHR #1 form **7311** and FormTitan cross HL7 platform integrated EHR #2 form **7321** were created can determine the matching FormTitan pdf mapping elements **7340** for example first name **7341**, last name **7342**, date of birth **7343** and allergies **7344**. The data from the existing matching records can be assigned to the FormTitan pdf mapping elements **7340** used to create the FormTitan cross HL7 platform integrated EHR #3 form **7332**.

[0395] The retrieval of matching EHR record data and automatic assignment and entry into corresponding Form-Titan pdf mapping elements **7340** of a FormTitan cross HL7 platform integrated EHR form saves time and potential typographic errors for and MD office. The EHR data can includes other patient information including driver license, address, SSN where applicable, other patient ID data, heath insurance provider and other information included in the patient EHR that the current MD needs to provide the patient with appropriate health care treatment of one embodiment.

#### Dropdown2 Form Settings Element:

[0396] FIG. 74 shows for illustrative purposes only an example of a dropdown2 form settings element of one embodiment. FIG. 74 shows a formbuilder webpage for adding a Form settings **7460** that can include a DropDown2 **7400** additional dropdown element. In this example the DropDown2 **7400** additional dropdown element includes a Lookup **7410** search element and a DropDown **7420** selection element. A form canvas is being set up to include a Col. 1 **7430**, Col. 2 **7440**, Col. 3 **7450** with 4 rows **7455** of one embodiment.

[0397] The foregoing has described the principles, embodiments and modes of operation of the present invention. However, the invention should not be construed as being limited to the particular embodiments discussed. The above described embodiments should be regarded as illustrative rather than restrictive, and it should be appreciated that variations may be made in those embodiments by workers skilled in the art without departing from the scope of the present invention as defined by the following claims.

**1-20.** (canceled)

**21.** An online form builder platform, comprising:  
a customizable application builder program coupled to a computer server, wherein the computer server has non-transitory computer-readable storage medium with a program including instructions that, when executed by at least one computer processor, causes the program to allow a cloud-based Internet application creation user to create at least one interactive website form using at least one component of a non-line by line coding visualization tool;

a database integration module coupled to a third party cloud-based relational database and configured to allow the application creation user to integrate real-time bi-directional transmission of data in the cloud-based relational database within a form element having conditional predetermined rules for transmitting the data to and from the interactive website form;

a customizable biometric tool configured to allow the application creation user to drag and drop biometric input field form elements of the visualization tool for the interactive website form, wherein the biometric input field form element allows the end user to input biometric data in the interactive website form to provide login security with validation of human related metrics;

a dynamic mapping tool configured to allow the application creation user to drag and drop input form elements of the visualization tool associated with the bi-directional data and input from the end user of the interactive website form to dynamically generate a customized portable document format file; and

a manual mapping tool configured to allow the application creation user to drag and drop the input form elements of the visualization tool on a predetermined portable document format template file, wherein a customize portable document format is generated with the bi-directional data and the input from the user of the interactive website form.

**22.** The online form builder platform of claim **21**, wherein the biometric input field form elements include a fingerprint recognition element configured to identify a user.

**23.** The online form builder platform of claim **21**, wherein the biometric input field form elements include an iris eye scan and full facial recognition configured to identify a user.

**24.** The online form builder platform of claim **21**, wherein the manual mapping tool predetermined portable document format template file is configured for overlaying interactive customize portable document format mapping elements onto an application creation user existing customize portable document format form to create an html interactive customize portable document format form.

**25.** The online form builder platform of claim **21**, wherein the customizable application builder program includes a push/pull feature configured to automatically save the data entered in the at least one interactive website form and simultaneously to the third party cloud-based relational database without the user having to enter any additional commands.

**26.** The online form builder platform of claim **21**, wherein the dynamic mapping tool is configured to convert a user's scanned existing paper form into a responsive customized portable document format template.

**27.** The online form builder platform of claim **21**, wherein the dynamic mapping tool includes an auto portable document format feature configured to create a portable document format of a hypertext markup language representation of the interactive form with the input data in the interactive form.

**28.** An apparatus, comprising:

a customizable application builder program coupled to a computer server, wherein the computer server has non-transitory computer-readable storage medium with a program including instructions that, when executed by at least one computer processor, causes the program

to allow a cloud-based Internet application creation user to create at least one interactive website form using at least one drag and drop form element of a visualization tool;  
a database integration module coupled to a third party cloud-based relational database and configured to allow the application creation user to integrate real-time bi-directional transmission of data in the cloud-based relational database within a form element;  
a dynamic mapping tool configured to allow the application creation user to dynamically generate a customized portable document format file having drag and drop input form elements of the visualization tool associated with the bi-directional data and input from a user of the interactive website form; and  
a manual mapping tool configured to allow the application creation user to drag and drop the input form elements of the visualization tool on a predetermined portable document format template file.

**29.** The apparatus of claim 28, wherein the dynamic mapping tool is configured to convert a users scanned existing paper form into a responsive customized portable document format file form.

**30.** The apparatus of claim 28, wherein the predetermined portable document format template file is configured to generate a customized portable document format with the bi-directional data and the input from the user of the interactive website form.

**31.** The apparatus of claim 28, wherein the customizable application builder program includes a push/pull feature configured to automatically save the data entered in the at least one interactive website form and simultaneously to the third party cloud-based relational database without the user having to enter any additional commands.

**32.** The apparatus of claim 28, wherein the manual mapping tool predetermined portable document format template file is configured for overlaying interactive customize portable document format mapping elements onto an application creation user existing customize portable document format form to create an html interactive customize portable document format.

**33.** The apparatus of claim 28, further comprising a customizable biometric tool configured to allow the application creation user to drag and drop biometric input field form elements of the visualization tool for the interactive website form, wherein the biometric input field form element allows an end user to input biometric data in the interactive website form to provide login security with validation of human related metrics.

**34.** The apparatus of claim 28, wherein the cloud-based relational database is configured for having conditional predetermined rules for transmitting the data to and from the interactive website form.

**35.** A method, comprising:

providing a customizable application builder program coupled to an Internet cloud-connected computer server for allowing an application creation user to create at least one interactive website form using at least one component of a non-line by line coding visualization tool;

using a database integration module coupled to a third party cloud-based relational database and to allow the

application creation user to integrate real-time bi-directional transmission of data in the cloud-based relational database within a form element having conditional predetermined rules for transmitting the data to and from the interactive website form;

providing a customizable smart validation element for allowing at least one end user with at least one login for accessing information associated with the integrated bi-directional form data of the third party cloud-based relational database that is an alternative login to an administrative login of the third party cloud-based relational database;

providing a dynamic mapping tool coupled to the customizable application builder program to allow the application creation user to drag and drop input form elements of the visualization tool associated with the bi-directional data and input from the end user of the interactive website form to dynamically generate a customized portable document format file; and

providing a manual mapping tool coupled to the customizable application builder program to allow the application creation user to drag and drop the input form elements of the visualization tool on a predetermined portable document format template file, wherein a customize portable document format is generated with the bi-directional data and the input from the end user of the interactive website form.

**36.** The method of claim 35, wherein providing the manual mapping tool includes providing interactive customize portable document format mapping elements to create an html interactive customize portable document format form.

**37.** The method of claim 35, wherein providing the customizable application builder program includes providing a push/pull feature to automatically push data entered by an end user from the at least one interactive website form to the third party cloud-based relational database and to automatically pull data from the third party cloud-based relational database and enter it into predetermined fields of the at least one interactive website form.

**38.** The method of claim 35, wherein providing the dynamic mapping tool includes providing an auto portable document format feature to create a portable document format of a hypertext markup language representation of the interactive form with the input data in the interactive form.

**39.** The method of claim 35, wherein providing the dynamic mapping tool includes providing a converter to convert an image into a responsive customized portable document format template.

**40.** The method of claim 35, further comprising providing a customizable biometric tool to allow the application creation user to drag and drop biometric input field form elements of the visualization tool for the interactive website form, wherein the biometric input field form element allows the end user to input biometric data in the interactive website form to provide login security with validation of human related metrics.