United States Patent Application Publication	20250258997
Kind Code	A1
Publication Date	August 14, 2025
Inventor(s)	Yousaf; Timothy et al.

COLLABORATIVE SPREADSHEET DATA VALIDATION AND INTEGRATION

Abstract

A system comprising a computer-readable storage medium storing at least one program and a method for integrating collaborative spreadsheet data into one or more network applications is presented. The method may include accessing an application data schema comprising a set of constraints on application data consumed by an application hosted by an application server. The method may further include accessing a spreadsheet having one or more data validation rules. The method may further include determining whether the one or more data validation rules include the set of constraints. In response to determining the one or more data validation rules include the set of constraints, application data consumed by the application is synchronized with spreadsheet data corresponding to the spreadsheet.

Inventors: **Yousaf; Timothy (New York, NY), Xing; Yichen (Potomac, MD), Brett-Bowen; Rhys (London, GB)**

Applicant: **Palantir Technologies Inc.** (Denver, CO)

Family ID: **59564117**

Appl. No.: **19/063544**

Filed: **February 26, 2025**

Related U.S. Application Data

parent US continuation 17664332 20220520 parent-grant-document US 12271686 child US 19063544
parent US continuation 16541524 20190815 parent-grant-document US 11366959 child US 17664332

Publication Classification

Int. Cl.: **G06F40/18** (20200101); **G06F3/0481** (20220101); **G06F3/14** (20060101); **G06F16/27** (20190101); **G06F40/174** (20200101); **G06F40/197** (20200101); **G06F40/226** (20200101); **H04L67/1095** (20220101)

U.S. Cl.:

CPC **G06F40/18** (20200101); **G06F3/14** (20130101); **G06F16/273** (20190101); **G06F40/174** (20200101); **G06F40/226** (20200101); **H04L67/1095** (20130101); G06F3/0481 (20130101); G06F40/197 (20200101)

Background/Summary

PRIORITY APPLICATION [0001] This application claims priority to U.S. Provisional Application Ser. No. 62/373,615, filed Aug. 11, 2016, the disclosure of which is incorporated herein in its entirety by reference.

TECHNICAL FIELD
[0002] The present disclosure generally relates to the technical field of special-purpose machines that facilitate integration of spreadsheet data including computerized variants of such special-purpose machines and improvements to such variants, and to the technologies by which such special-purpose machines become improved compared to other special-purpose machines that facilitate integration of spreadsheet data. In particular, the present disclosure addresses systems and methods for collaborative spreadsheet data entry, validation, and integration into one or more network applications.
BACKGROUND
[0003] Traditional spreadsheet applications such as Microsoft® Excel® are frequently used to collect many different kinds of data as they are convenient tools for ad hoc data-entry. However, such traditional spreadsheet tools are not conducive for use in collaborative environments. For example, with traditional spreadsheet tools it is difficult for multiple users to collaborate on a single spreadsheet because if one user has the spreadsheet open, other users will be unable to open it. Another problem with traditional spreadsheets occurs in scenarios where multiple people are collaborating on multiple spreadsheets and the data schema used varies across groups and across spreadsheets. This may create an issue if the spreadsheets are being merged because a user must manually munge the spreadsheets together so they are in a uniform schema. An additional problem with traditional spreadsheets is that if data is being gathered using a spreadsheet and it is gathered to perform an analysis, the analysis is likely to be out of date because the data is just a snapshot from when it is collected. Once the spreadsheet is updated in the field, a delay is introduced in the analysis because the user tasked with performing the analysis must wait until the updated data is collected to make the analysis “live.” Meanwhile, the data may again be updated in the field thereby rendering the analysis to be outdated.

Description

BRIEF DESCRIPTION OF THE DRAWINGS
[0004] Various ones of the appended drawings merely illustrate example embodiments of the present inventive subject matter and cannot be considered as limiting its scope.
[0005] FIG. **1** is a network diagram illustrating a network system comprising a collaboration platform configured to integrate spreadsheet data with a network-based application, according to some embodiments.

[0006] FIG. **2** is an architecture diagram illustrating a structure of the network system, according to some embodiments.
[0007] FIG. **3** is a system diagram illustrating various functional components of a spreadsheet application, which is provided as part of the collaboration platform, according to some embodiments.
[0008] FIG. **4** is a data structure diagram illustrating a spreadsheet data object, according to some embodiments.
[0009] FIGS. **5-6** are flowcharts illustrating a method for validating a spreadsheet, according to some embodiments.
[0010] FIG. **7** is a flowchart illustrating a method for integrating spreadsheet data with a network-based application, according to some embodiments.
[0011] FIGS. **8**A-**8**C, **9-10**, **11**A-**11**F, and **12-13** are interface diagrams illustrating various aspects of user interfaces provided by the collaboration platform, according to some embodiments.
[0012] FIG. **14** is a diagrammatic representation of a machine in the example form of a computer system within which a set of instructions for causing the machine to perform any one or more of the methodologies discussed herein may be executed.

DETAILED DESCRIPTION

[0013] Reference will now be made in detail to specific example embodiments for carrying out the inventive subject matter. Examples of these specific embodiments are illustrated in the accompanying drawings, and specific details are set forth in the following description in order to provide a thorough understanding of the subject matter. It will be understood that these examples are not intended to limit the scope of the claims to the illustrated embodiments. On the contrary, they are intended to cover such alternatives, modifications, and equivalents as may be included within the scope of the disclosure.
[0014] As noted above, with traditional spreadsheet applications it is difficult for multiple users to collaborate on a single file (e.g., a spreadsheet), it is difficult to enforce “clean” data entry, and the periodic analysis quickly becomes outdated. To address the foregoing problems (among others) with traditional systems, aspects of the present disclosure include a system and methods for collaborative spreadsheet data entry, validation, and integration into one or more network applications. The inventive subject matter may find particular application in workflows where multiple users continually record data in spreadsheets and periodically create an analysis from this data, although it shall be appreciated that the inventive subject matter is not limited to application in such workflows, and the inventive subject matter may find equal application in other workflows and scenarios.
[0015] With reference to the live collaboration aspect, multiple users may edit a single spreadsheet simultaneously. With reference to data validation, users may create schemas with one or more validation rules that the system enforces to ensure clean data entry. For example, as a rule, a user may specify that a particular column in a spreadsheet includes dates, and the system verifies that data enter into the column includes a valid date. As part of the data validation feature, a helper widget may be provided to assist users in entering data that conforms to the specified schema. Following the above example, the helper widget for a date column may include a date picker to ensure users enter a date into the column. Additionally, the data validation aspect provides a guarantee for the schema on the backend (e.g., the backend knows the column has a date).
[0016] With reference to the data integration aspect, data entered into the spreadsheet may be easily integrated into other applications and services for live analysis. In this way, the data used by the other applications will be the most recently entered data. Further, spreadsheet data will only be synchronized with applications if the spreadsheet schema matches the schema used by the corresponding application. In instances in which the two schemas do not match, users may add additional rules to the spreadsheet to ensure compliance and proper synchronization.
[0017] In an example, the system allows users to collaborate on a spreadsheet to record locations

they want to see plotted on a map using a network-based mapping application. In this example, the system may validate data entered into the spreadsheet (e.g., locations) against a set of validation rules that corresponds to a data schema used by the network-based mapping application and, responsive to determining the set of validation rules matches an application data schema used by the network-based map application, the system synchronizes spreadsheet data with application data consumed by the network-based mapping application thereby enabling the users to view locations plotted on a map by the network-based map application as users are entering the locations into the spreadsheet.

[0018] FIG. **1** is a network diagram depicting a network system **100** comprising a collaboration platform **102** configured to integrate spreadsheet data **104** with a network-based application **106**, according to some embodiments. As shown, the network system **100** includes the collaboration platform **102**, an application server **108** hosting the network-based application **106**, and user devices **110**A-C, all communicatively coupled to each other via a network **112**. In an example, the collaboration platform **102** shown in FIG. **1** employs a client-server architecture to exchange data with the user devices **110**A-C, although the present inventive subject matter is, of course, not limited to such an architecture, and could equally well find application in an event-driven, distributed, or peer-to-peer architecture system, for example. Moreover, it shall be appreciated that although some of the functional components of the network system **100** are discussed in the singular sense, multiple instances of one or more of the various functional components may be employed.

[0019] Also shown in FIG. **1** are users **114**A-C, who may be human users (e.g., human beings), machine users (e.g., computers configured by a software program to interact with the device **110**), or any suitable combination thereof (e.g., a human assisted by a machine or a machine supervised by a human). The users **114**A-C are respectively associated with the user devices **110**A-C and may be users of such devices. For example, the user devices **110**A-C may be any one of a desktop computer, a tablet computer, a smart phone, or a wearable device (e.g., a smart watch or smart glasses) belonging to any one of the users **114**A-C.

[0020] The user devices **110**A-C may also include any one of a web client **116** (e.g., a web browser) or application **118** to facilitate communication and interaction between the user device **110** and the collaboration platform **102**. In various embodiments, information communicated between the collaboration platform **102** and the user device **110** may involve user-selected functions available through one or more user interfaces (UIs). Accordingly, during a communication session with any one of the user devices **110**A-C, the collaboration platform **102** may provide a set of machine-readable instructions that, when interpreted by the user devices **110**A-C using the web client **116** or the application **118**, cause the user devices **110**A-C to present the UI, and transmit user input received through such a UI back to the collaboration platform **102**.

[0021] The collaboration platform **102** may be implemented in a special-purpose (e.g., specialized) computer system, in whole or in part, as described below. The collaboration platform **102** includes a spreadsheet application **120** designed for continual collection of human-created data. More specifically, the spreadsheet application **120** includes a front end that allows the users **114**A-C to interact with a spreadsheet **122** using either the web client **116** or the application **118**, and a backend that drives a view of the spreadsheet **122** and maintains a canonical version of the spreadsheet data **104** that is created. In this example, the spreadsheet **122** is a live representation of the spreadsheet data **104** maintained by the spreadsheet application **120**. The spreadsheet data **104** is stored in a data store (e.g., a computer-readable storage device) that forms part of, or is communicatively coupled to, the collaboration platform **102**. The collaboration platform **102** may further maintain one or more historical versions of the spreadsheet data **104** to enable the users **114**A-C to restore the spreadsheet **122** to a prior version.

[0022] Additionally, the spreadsheet application **120** allows the users **114**A-C to create validation rules associated with the spreadsheet **122**. Each validation rule includes a constraint that limits or

controls what the users **114** can enter into at least one cell of the spreadsheet **122**, and each is typically, but not necessarily always, associated with entire columns in the spreadsheet **122**.
[0023] The spreadsheet application **120** is also responsible for synchronizing the spreadsheet data **104** with application data **124** consumed (e.g., used) by the application **106** hosted by the application server **108**. The application **106** may configure the application server **108** to provide any number of network-based services that consume application data **124** to provide data manipulation, presentation, communication, or other capabilities to the users **114**A-C or other users.
[0024] The network **112** may be any network that enables communication between or among systems, machines, databases, and devices (e.g., between collaboration platform **102** and the devices **110**A-C). Accordingly, the network **112** may be a wired network, a wireless network (e.g., a mobile or cellular network), or any suitable combination thereof. The network **112** may include one or more portions that constitute a private network, a public network (e.g., the Internet), or any suitable combination thereof. Accordingly, the network **112** may include one or more portions that incorporate a local area network (LAN), a wide area network (WAN), the Internet, a mobile telephone network (e.g., a cellular network), a wired telephone network (e.g., a plain old telephone system (POTS) network), a wireless data network (e.g., a WiFi network or WiMax network), or any suitable combination thereof. Any one or more portions of the network **112** may communicate information via a transmission medium. As used herein, “transmission medium” refers to any intangible (e.g., transitory) medium that is capable of communicating (e.g., transmitting) instructions for execution by a machine (e.g., by one or more processors of such a machine), and includes digital or analog communication signals or other intangible media to facilitate communication of such software.
[0025] FIG. **2** is an architecture diagram illustrating a structure of the network system **100**, according to some embodiments. As shown in FIG. **2**, the collaboration platform **102** and the application **106** may interact, via an application programming interface (API) **202**, with a representational state transfer (REST) server **204** to synchronize the spreadsheet data **104** with the application data **124** consumed by the application **106** (e.g., application data **124**). For example, the spreadsheet application **120** may communicate, via the API **202**, with the REST server **204** to integrate the spreadsheet data **104** with data sources **206** that include one or more data repositories (e.g., databases) that provide data to the application **106**. Likewise, the application **106** may communicate, via the API **202**, with the REST server **204** to integrate changes to the application data **124** with the data sources **206**, which may, in turn, be consumed by the spreadsheet application **120**. Additionally, the collaboration platform **102** and the application **106** are in communication with one or more components **208** that provide additional functionality to the collaboration platform **102** and the application **106** related to data included in the data sources **206**. For example, the one or more components **208** may provide a data object viewer, a document viewer, search templates, ontology chooser, or an investigation bar to the users **114**A-C of either the spreadsheet application **120** or the application **106**.
[0026] FIG. **3** is a system diagram illustrating various functional components of the spreadsheet application **120**, according to some embodiments. To avoid obscuring the inventive subject matter with unnecessary detail, various functional components (e.g., modules, engines, and databases) that are not germane to conveying an understanding of the inventive subject matter have been omitted from FIG. **3**. However, a skilled artisan will readily recognize that various additional functional components may be supported by the spreadsheet application **120** to facilitate additional functionality that is not specifically described herein. As shown, the spreadsheet application **120** includes: an interface module **300**; a rule management module **302**; a validation module **304**; and a synchronization module **306**. Each of the above referenced functional components of the spreadsheet application **120** are configured to communicate with each other (e.g., via a bus, shared memory, a switch, or application programming interfaces (APIs) **202**).

[0027] The interface module **300** receives requests from the user devices **110**A-C, and communicates appropriate responses to the user devices **110**A-C. The interface module **300** may receive requests from devices in the form of hypertext transfer protocol (HTTP) requests or other web-based, API requests. For example, the interface module **300** provides a number of interfaces (e.g., APIs **202**) that allow data to be exchanged between the user devices **110**A-C and the collaboration platform **102**.
[0028] The interface module **300** also provides UIs to the user devices **110**A-C that allow the users **114**A-C to view and interact with the spreadsheet **122**. To provide a UI to one of the user devices **110**A-C, the interface module **300** transmits a set of machine-readable instructions to the user device **110** that causes the user device **110** to present the UI on a display of the user device **110**. The set of machine-readable instructions may, for example, include presentation data (e.g., representing various elements of the UI), the spreadsheet data **104**, and a set of instructions to display the presentation data. The receiving device (e.g., one of the user devices **110**A-C) may temporarily store the presentation data and the spreadsheet data **104** to enable display of the UI and interaction with the spreadsheet **122** from within the UI.
[0029] The UIs provided by the interface module **300** may also include various input control elements (e.g., sliders, buttons, drop-down menus, check-boxes, and data entry fields) that allow the users **114**A-C to specify various inputs such as updates to cells of the spreadsheet **122** or validation rules associated with the spreadsheet **122**. The interface module **300** receives and processes user input received through such input control elements, and in some instances, the interface module **300** may update the spreadsheet data **104** in accordance with the received input (e.g., the interface module **300** updates the spreadsheet data **104** in accordance with edits made to the spreadsheet **122** by any one of the users **114**A-C). Examples of the UIs provided by the interface module **300** are discussed below in reference to FIGS. **8**-**13**.
[0030] The rule management module **302** is responsible for managing validation rules associated with the spreadsheet **122**. Each validation rule includes a constraint on data entered into at least one cell of the spreadsheet **122**, and in many cases, on all cells of an entire column. For example, a validation rule may constrain data entered into a column specifically to dates. In this example, any non-date entry entered into a cell of the spreadsheet will not conform to the validation rule.
[0031] The rule management module **302** allows the users **114**A-C to add, edit, or delete validation rules associated with the spreadsheet **122**. To this end, the rule management module **302** may work in conjunction with the interface module **300** to provide a rule management interface that allows the users **114**A-C to view, add, edit, or delete validation rules. An example rule management interface is illustrated in FIGS. **11**A-**11**F and described in further detail below, in accordance with some embodiments. Further, as will be described below in reference to FIG. **4**, the rule management module **302** stores validation rules as part of a spreadsheet artifact, which corresponds to the spreadsheet data **104**.
[0032] The validation module **304** is configured to validate the spreadsheet **122** according to one or more validation rules associated with the spreadsheet **122**. In validating the spreadsheet **122**, the validation module **304** compares information entered into each cell with any validation rules associated with the cell to determine whether the information entered into the cell conforms to the associated validation rules. In an example, a validation rule associated with the spreadsheet **122** specifies that only dates should be entered into a particular column. In this example, the validation module **304** checks whether the information entered into each cell of the column is a date.
[0033] Cells with entries that conform to the validation rules are considered to include valid entries, whereas cells with entries that do not conform to the validation rules are considered to include invalid entries. In response to determining that a cell includes an invalid entry, the validation module **304** works in conjunction with the interface module **300** to cause the cell with the invalid entry to be visually distinguished from cells with valid entries. For example, the cell with the invalid entry may be highlighted or otherwise displayed differently (e.g., in a different color, font,

format, or style) from entries in the spreadsheet **122** that include valid entries. An example of how the interface module **300** visually distinguishes a cell is illustrated in FIG. **12** and discussed below in further detail in accordance with some embodiments.

[0034] The synchronization module **306** is configured to synchronize the spreadsheet data **104** and the application data **124**. That is, the synchronization module **306** is responsible for ensuring that changes made to the spreadsheet data **104** are reflected in the application data **124**, and that changes made to the application data **124** are reflected in the spreadsheet data **104**. In synchronizing the spreadsheet data **104** with the application data **124**, the synchronization module **306** may communicate, via the API **202**, one or more requests to the REST server **204** to integrate the spreadsheet data **104** and any subsequent changes made thereto with the data sources **208** that supply the application data **124** to the application **106**.

[0035] In many instances, the application **106** employs a particular application data schema that includes a set of constraints on the application data **124**. For example, the application data schema for a map application may specify that the application data **124** be in the form of geo-coordinates (e.g., latitude and longitude, or military grid reference system (MGRS)). In these instances, the synchronization module **306** ensures that the validation rules associated with the spreadsheet **122** match the application data schema for the application **106** prior to synchronizing the spreadsheet data **104** with the application data **124**. In other words, the synchronization module **306** compares the validation rules associated with the spreadsheet **122** to the application data schema to determine whether the validation rules include the set of constraints that are included in the application data schema. In this way, the synchronization module **306** ensures that the application **106** is not provided with invalid values that may lead to errors or other issues.

[0036] As is understood by skilled artisans in the relevant computer and Internet-related arts, each functional component (e.g., engine, module, or database) illustrated in FIG. **3** may be implemented using hardware (e.g., a processor of a machine) or a combination of logic (e.g., executable software instructions) and hardware (e.g., memory and processor of a machine) for executing the logic. Furthermore, the various functional components depicted in FIG. **3** may reside on a single machine (e.g., a server) or may be distributed across several machines in various arrangements such as cloud-based architectures. Moreover, any two or more of these components may be combined into a single component (e.g., a single module), and the functions described herein for a single component may be subdivided among multiple modules. Functional details of these modules are described below with respect to FIGS. **5-8**.

[0037] FIG. **4** is a data structure diagram illustrating a spreadsheet artifact **400**, according to some embodiments. The spreadsheet artifact **400** is a tree data structure used by the collaboration platform **102** to represent spreadsheets. In an example, the spreadsheet artifact **400** is used to represent the spreadsheet **122** and corresponds to the spreadsheet data **104**. As shown, the spreadsheet artifact **400** includes four different non-primitive data types that make up a spreadsheet state to be stored and shared. The element types include: sheet **402**, column **404**, cell **406**, and column type **408**. The sheet **402** includes a name, a creator name, and a list of columns in the spreadsheet. The column **404** includes a name and enumeration for type (e.g., date, text, file) along with one or more validation rules and a list of cells in the column. The cell **406** includes a value and lock information. The column type **408** includes an enumeration for type along with additional configuration information. Consistent with some embodiments, the column type **408** may be used for client side data validation.

[0038] FIGS. **5-6** are flowcharts illustrating a method **500** for validating a spreadsheet, according to some embodiments. The method **500** may be embodied in computer-readable instructions for execution by one or more processors such that the operations of the method **500** may be performed in part or in whole by the collaboration platform **102**; accordingly, the method **500** is described below by way of example with reference thereto. However, it shall be appreciated that at least some of the operations of the method **500** may be deployed on various other hardware configurations and

the method **500** is not intended to be limited to the collaboration platform **102**.
[0039] At operation **505**, the rules management module **302** receives a validation rule associated with the spreadsheet **122**. The validation rule may be specified by any one of the users **114**A-C using a UI provided to one of the user devices **110**A-C by the interface module **300**. As noted above, the validation rule imposes a constraint on information the users **114**A-C can enter into the cells **406** of a specific column of the spreadsheet **122**.
[0040] The validation rule comprises validation logic and, optionally, one or more configuration parameters. The validation logic includes a name, a validator, and a value type. The interface module **300** provides user interface elements that allow the users **114**A-C to specify the validation logic (e.g., name, validator, and value type) and the one or more configuration parameters. Accordingly, the receiving of the validation rule may include receiving user specified validation logic, and receiving one or more user specified configuration parameters.
[0041] At operation **510**, the rules management module **302** stores the validation rule (e.g., in a computer-readable storage device forming part of or coupled to the collaboration platform **102**) in association with the spreadsheet **122**. In particular, the rules management module **302** stores the validation rule as part of the corresponding column **404** forming part of the spreadsheet artifact **400**.
[0042] At operation **515**, the validation module **304** validates the spreadsheet **122** according to validation rules associated with the spreadsheet **122** (e.g., the validation rule received at operation **505**). In validating the spreadsheet **122** according to the validation rules, the validation module **304** checks each cell **406** to ensure that the value entered into the cell complies with a validation rule associated with the cell (e.g., a validation rule associated with the column in which the cell resides).
[0043] A value that does not comply with a corresponding validation rule is an invalid value. At operation **520**, the validation module **304** determines whether the spreadsheet **122** includes a cell with an invalid value. If, at operation **520**, the validation module **304** determines the spreadsheet **122** includes a cell with an invalid value, the method **500** proceeds to operation **520**, where the validation module **304** works with the interface module **300** to cause the cell with the invalid value to be visually distinguished (e.g., highlight) compared to the of remainder of cells included in the presentation of the spreadsheet **122**. If, at operation **520**, the validation module **304** determines all cells include valid entries, the method **500** ends.
[0044] As shown in FIG. **6**, the method **500** may include one or more of operations **605** and **610**. As shown in FIG. **6**, operation **605** may be performed after operation **505**, in which the rules management module **302** receives a validation rule. At operation **605**, the interface module **300** provides a help widget for display on the user devices **110**A-C. The help widget comprises one or more input elements operable to enter values into one or more cells (e.g., cells of a particular column) through appropriate user interaction (e.g., a mouse click). For example, the help widget may be or include a date picker that allows the users **114**A-C to select a date from an input element that resembles a calendar. In another example, the help widget may be or include a drop-down menu with a list of predefined selectable values. In yet another example, the help widget may be or include a geo-coordinate widget that allows users **114**A-C to input a set of geo-coordinates (e.g., a longitude and latitude).
[0045] In providing the help widget, the interface module **300** may provide a set of computer-readable instructions to the user devices **110**A-C that configure the user devices **110**A-C to display and receive input from the help widget. As implied by the above referenced examples, the interface module **300** may be configured to provide various types of help widgets for inputting various types of information, and the type of help widget provided may be based on the validator included in the validation rule. For example, the interface module **300** may provide the date picker in response to the validation rule including a date validator, or a geo-coordinate widget in response to the validation rule including a coordinate validator.
[0046] At operation **610**, the interface module **300** receives an update to one or more cells of the

spreadsheet **122**. The update may include a new value, a deleted value, or a modification (e.g., edit) to an existing value. The update may be made by one or more of the users **114**A-C using the user devices **110**A-C to either manually enter the update or to enter the update using the help widget. In an example, the update may include: a modification (e.g., a change to a value) made to a first cell of the spreadsheet **122** by the user **114**A using the user device **110**A; a first new value entered into a second cell of the spreadsheet **122** by the user **114**B using the user device **110**B; and a second new value entered into a third cell of the spreadsheet **122** by the user **114**C using the user device **110**C.

[0047] As shown in FIG. **6**, operation **610** may be performed prior to operation **510**, in which the validation module **304** validates the spreadsheet **122**. Accordingly, the validating of the spreadsheet **122** in operation **510** may include validating the update to the one or more cells (e.g., new values or modified values) received in operation **610** according to the one or more validation rules associated with the spreadsheet **122**.

[0048] FIG. **7** is a flowchart illustrating a method **700** for integrating the spreadsheet data **104** with the application **106**, according to some embodiments. The method **700** may be embodied in computer-readable instructions for execution by one or more processors such that the operations of the method **700** may be performed in part or in whole by the collaboration platform **102**; accordingly, the method **700** is described below by way of example with reference thereto. However, it shall be appreciated that at least some of the operations of the method **700** may be deployed on various other hardware configurations and the method **700** is not intended to be limited to the collaboration platform **102**. In some embodiments, the method **700** may be performed subsequent to the method **500**, during which the collaboration platform **102** validates the spreadsheet data **104** and visually distinguishes any invalid entries in the spreadsheet **122**.

[0049] At operation **705**, the synchronization module **306** accesses an application data schema corresponding to the application **106** executing on the application server **108**. The application data schema includes a set of constraints to control or limit the application data **124** consumed by the application **106** (e.g., input data provided to the application **106**). In an example, the set of constraints limits the application data **124** to dates input in a particular form (e.g., MM/DD/YYYY). In another example, the set of constraints limits the application data **124** to geo-location coordinates (e.g., longitude and latitude or MSGR).

[0050] At operation **710**, the synchronization module **306** accesses one or more validation rules associated with the spreadsheet **122**. As noted above, the validation rules include constraints that limit or control information entered into the spreadsheet **122**. The one or more validation rules may, for example, include the validation rule received at operation **505**. In accessing the one or more validation rules, the synchronization module **306** may access the spreadsheet artifact **400** and identify the one or more validation rules from one or more columns **404** included in the spreadsheet artifact **400**.

[0051] At operation **715**, the synchronization module **306** determines whether the one or more validation rules associated with the spreadsheet **122** include the set of constraints included in the application data schema of the application **106**. For example, if the application data schema limits the application data **124** consumed by the application **106** to dates, the synchronization module **306** checks one or more validation rules to determine whether they include a date validation rule limiting values entered into at least one column in the spreadsheet **122** to dates.

[0052] If, at operation **715**, the synchronization module **306** determines the one or more validation rules associated with the spreadsheet **122** include the set of constraints, the synchronization module **306** synchronizes the spreadsheet data **104** with the application data **124**, at operation **720**. For example, the synchronization module **306** may communicate, via the API **202**, one or more requests to the REST server **204** to integrate the spreadsheet data **104** and any subsequent updates made thereto (e.g., at operation **610**) with the data sources **208** that supply the application data **124** to the application **106**.

[0053] If, at operation **720**, the synchronization module **306** determines the one or more validation

rules associated with the spreadsheet **122** do not include the set of constraints, the spreadsheet data **104** is not synchronized with the application data **124**, and the synchronization module **306** works in conjunction with the interface module **300** to cause an alert to be displayed within an interface element presented on at least one of the user devices **110**A-C, at operation **725**. The alert may include a notification that the spreadsheet data **104** and the application data **124** are not being synchronized, and may further list any additional validation rules that should be associated with the spreadsheet **122**.

[0054] FIGS. **8-13** are interface diagrams illustrating various aspects of user interfaces provided by the collaboration platform **102**, according to some embodiments. More specifically, each of the user interfaces illustrated in FIGS. **8-13** and discussed below may be provided by the interface module **300** and presented on a display unit of any of the user devices **110**A-C. Further, the users **114**A-C may interact with the user interfaces illustrated in FIGS. **8-13** as discussed below to perform live collaborations on spreadsheets and otherwise interact with the spreadsheet application **120** included as part of the collaboration platform **102**.

[0055] FIGS. **8**A-**8**C are interface diagrams illustrating multiple views of an example spreadsheet **800**, according to some embodiments. FIG. **8**A illustrates a message box **802** being presented in conjunction with the spreadsheet **800**. The message box **802** includes a list of users collaborating on the spreadsheet **800** (e.g., accessing and editing the spreadsheet **800**). The list of users may include or correspond to the users **114**A-C. Users may access the message box **802** through appropriate interaction with the button **804** (e.g., through selection using a mouse cursor).

[0056] FIG. **8**B illustrates a version history dialog box **806** presented in conjunction with the spreadsheet **800**. The version history dialog box **806** includes a list of historical versions (e.g., prior version) of the spreadsheet **800**. A user selection of any one of the prior versions of the spreadsheet **800** causes the spreadsheet **800** to be reverted to the selected prior version of the spreadsheet **800**. Users may access the version history dialog box **806** through appropriate interaction with a button **808** (e.g., through selection using a mouse cursor).

[0057] FIG. **8**C illustrates a synchronization dialog box **810** presented in conjunction with the spreadsheet **800**. The synchronization dialog box **810** presents information related to the synchronization of spreadsheet data corresponding to the spreadsheet **800** with application data of one or more network-based applications. In this example, the synchronization dialog box **810** includes information related to synchronization of the spreadsheet data with a map application. In particular, the synchronization dialog box **810** includes an alert with a notification that the spreadsheet **800** has not been synchronized with the application data of the map application, which, as discussed above, may be presented in response to determining the validation rules of the spreadsheet **800** do not include the application data schema of the map application. The synchronization dialog box **810** further includes a list of constraints included in the application data schema of the map application that may be included in the validation rules of the spreadsheet **800** to trigger synchronization.

[0058] The synchronization dialog box **810** also includes a toggle **812** that allows users to authorize the synchronization of the spreadsheet data of the spreadsheet **800** with application data of network-based applications. Further, users may access the synchronization dialog box **810** through appropriate interaction with the button **814** (e.g., through selection using a mouse cursor).

[0059] FIG. **9** is an interface diagram illustrating a portion of an example spreadsheet **900**, according to some embodiments. As shown, the name field of each column (e.g., the top most row) includes a button **902** that users may select to access a rules management interface from which users can view, edit, or add validation rules associated with the column. An example rules management interface is illustrated in FIGS. **11**A-**11**F and described below in accordance with some embodiments.

[0060] FIG. **10** is an interface diagram illustrating a column creation window **1000**, according to some embodiments. The column creation window **1000** may be used to add a column to a

spreadsheet and to configure one or more validation rules for the column. For example, the drop-down list **1002** provides users with a list of predefined validation rules to select from. Additionally, the column creation window **1000** includes a button **1004** that users may select to access a rules management interface from which users can view, edit, or add validation rules associated with the column. An example rules management interface is illustrated in FIGS. **11**A-**11**F and described below in accordance with some embodiments.

[0061] FIGS. **11**A-**11**F are interface diagrams illustrating a rules management interface **1100** configured for viewing, editing, and creating validation rules associated with a spreadsheet (e.g., spreadsheet **122** or **800**), according to some embodiments. As shown, in FIG. **11**A the rules management interface **1100** includes a list box **1102** comprising a list of predefined validation rules from which users may select a validation rule to view and edit the validation logic along with any configuration parameters. Accordingly, the rules management interface **1100** includes a number of input controls (e.g., selectable drop-down menus and text input fields) that allow users to edit the validation logic (e.g., name, validator, format, and value type) of the validation rule. More specifically, a user (e.g., one of the users **114**A-C) may use: text field **1104** to enter a name for the validation rule; drop-down menu **1106** to enter a validator for the validation rule; text field **1108** to enter a format for values; and drop-down menu **1110** to specify a value type.

[0062] In the example illustrated in FIG. **11**A, a user (e.g., one of the users **114**A-C) has selected a rule **1112** that constrains data entered into cells of a particular column to dates. In the example illustrated in FIG. **11**B, a user has selected a rule **1114** that constrains data entered into cells of a particular column to a unique number. In the example illustrated in FIG. **11**C, a user has selected a rule **1116** that constrains data entered into cells of a particular column to a provinces in a particular number. As part of creating the rule **1116**, the user may specify an additional spreadsheet and a column from that spreadsheet from which the provinces may be selected. Further, while a user is editing values in the column, the interface module **300** may provide a drop-down list comprising the list of provinces extracted from the specified column of the additional spreadsheet.

[0063] As shown in FIG. **11**D, the list box **1102** also includes a selectable element **1118** that allows users to create a new validation rule. As part of this process, a user may specify a name for the new validation rule in text field **1120**. Further, as shown in FIG. **11**E, the user may select a validator for the validation rule using drop-down menu **1122**. Lastly, as shown in FIG. **11**F, the user may select a value type for the validation rule from the drop-down menu **1124**.

[0064] FIG. **12** is an interface diagram illustrating a portion of an example spreadsheet **1200**, according to some embodiments. As shown in FIG. **12**, the spreadsheet **1200** includes a cell **1202**, which has been visually distinguished (e.g., highlighted) from the remainder of the cells in the spreadsheet **1200**. The cell **1202** may be visually distinguished in this manner in response to the validation module **304** determining the cell includes an invalid value (e.g., a value that does not comply with the validation rule for that column). As shown, a notification **1204** is also provided to notify users that the value is invalid.

[0065] FIG. **13** is an interface diagram illustrating a synchronization interface **1300** configured to provide information related to synchronization of spreadsheet data with application data of network-based application **1302** (search application), network-based application **1304** (graphing application), network-based application **1306** (table creation application), and network-based application **1308** (map application), according to some embodiments. In particular, the synchronization interface **1300** provides a status of synchronization with the network-based applications **1302**, **1304**, **1306**, and **1308** (e.g., synched or not synched). In some instances, a network-based application (e.g., application **1308**) may employ a particular application data schema, and a list of constraints included in the application data schema of the application are presented to the user. Further, the synchronization module **306** may prevent spreadsheet data from being synchronized with application data if the validation rules of the corresponding spreadsheet do not include the constraints of the application data schema. Additionally, as with the synchronization

dialog box **810** discussed in reference to FIGS. **8**A-**8**C, the synchronization interface **1300** also includes a toggle **1310** that allows users to authorize the synchronization of the spreadsheet data of the spreadsheet with application data of network-based applications **1302**, **1304**, **1306**, and **1308**.

Modules, Components, and Logic

[0066] Certain embodiments are described herein as including logic or a number of components, modules, or mechanisms. Modules may constitute either software modules (e.g., code embodied on a machine-readable medium) or hardware modules. A "hardware module" is a tangible unit capable of performing certain operations and may be configured or arranged in a certain physical manner. In various example embodiments, one or more computer systems (e.g., a standalone computer system, a client computer system, or a server computer system) or one or more hardware modules of a computer system (e.g., a processor or a group of processors) may be configured by software (e.g., an application or application portion) as a hardware module that operates to perform certain operations as described herein.

[0067] In some embodiments, a hardware module may be implemented mechanically, electronically, or any suitable combination thereof. For example, a hardware module may include dedicated circuitry or logic that is permanently configured to perform certain operations. For example, a hardware module may be a special-purpose processor, such as a field-programmable gate array (FPGA) or an application specific integrated circuit (ASIC). A hardware module may also include programmable logic or circuitry that is temporarily configured by software to perform certain operations. For example, a hardware module may include software executed by a general-purpose processor or other programmable processor. Once configured by such software, hardware modules become specific machines (or specific components of a machine) uniquely tailored to perform the configured functions and are no longer general-purpose processors. It will be appreciated that the decision to implement a hardware module mechanically, in dedicated and permanently configured circuitry, or in temporarily configured circuitry (e.g., configured by software) may be driven by cost and time considerations.

[0068] Accordingly, the phrase "hardware module" should be understood to encompass a tangible entity, be that an entity that is physically constructed, permanently configured (e.g., hardwired), or temporarily configured (e.g., programmed) to operate in a certain manner or to perform certain operations described herein. As used herein, "hardware-implemented module" refers to a hardware module. Considering embodiments in which hardware modules are temporarily configured (e.g., programmed), each of the hardware modules need not be configured or instantiated at any one instance in time. For example, where a hardware module comprises a general-purpose processor configured by software to become a special-purpose processor, the general-purpose processor may be configured as respectively different special-purpose processors (e.g., comprising different hardware modules) at different times. Software accordingly configures a particular processor or processors, for example, to constitute a particular hardware module at one instance of time and to constitute a different hardware module at a different instance of time.

[0069] Hardware modules can provide information to, and receive information from, other hardware modules. Accordingly, the described hardware modules may be regarded as being communicatively coupled. Where multiple hardware modules exist contemporaneously, communications may be achieved through signal transmission (e.g., over appropriate circuits and buses) between or among two or more of the hardware modules. In embodiments in which multiple hardware modules are configured or instantiated at different times, communications between such hardware modules may be achieved, for example, through the storage and retrieval of information in memory structures to which the multiple hardware modules have access. For example, one hardware module may perform an operation and store the output of that operation in a memory device to which it is communicatively coupled. A further hardware module may then, at a later time, access the memory device to retrieve and process the stored output. Hardware modules may also initiate communications with input or output devices, and can operate on a resource (e.g., a

collection of information).
[0070] The various operations of example methods described herein may be performed, at least partially, by one or more processors that are temporarily configured (e.g., by software) or permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors may constitute processor-implemented modules that operate to perform one or more operations or functions described herein. As used herein, "processor-implemented module" refers to a hardware module implemented using one or more processors.
[0071] Similarly, the methods described herein may be at least partially processor-implemented, with a particular processor or processors being an example of hardware. For example, at least some of the operations of a method may be performed by one or more processors or processor-implemented modules. Moreover, the one or more processors may also operate to support performance of the relevant operations in a "cloud computing" environment or as a "software as a service" (SaaS). For example, at least some of the operations may be performed by a group of computers (as examples of machines including processors), with these operations being accessible via a network (e.g., the Internet) and via one or more appropriate interfaces (e.g., an API **202**).
[0072] The performance of certain of the operations may be distributed among the processors, not only residing within a single machine, but deployed across a number of machines. In some example embodiments, the processors or processor-implemented modules may be located in a single geographic location (e.g., within a home environment, an office environment, or a server farm). In other example embodiments, the processors or processor-implemented modules may be distributed across a number of geographic locations.
Example Machine Architecture
[0073] FIG. **14** is a block diagram illustrating components of a machine **1400**, according to some example embodiments, able to read instructions from a machine-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein. Specifically, FIG. **14** shows a diagrammatic representation of the machine **1400** in the example form of a computer system, within which instructions **1416** (e.g., software, a program, an application, an applet, an app, or other executable code) for causing the machine **1400** to perform any one or more of the methodologies discussed herein may be executed. Additionally, or alternatively, the machine **1400** may correspond to any one of the collaboration platform **102**, the user devices **110**A-C, or the application server **108**. The instructions transform the general, non-programmed machine into a particular machine programmed to carry out the described and illustrated functions in the manner described. In alternative embodiments, the machine **1400** operates as a standalone device or may be coupled (e.g., networked) to other machines. In a networked deployment, the machine **1400** may operate in the capacity of a server machine or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine **1400** may comprise, but not be limited to, a server computer, a client computer, a personal computer (PC), a tablet computer, a laptop computer, a netbook, a set-top box (STB), a PDA, an entertainment media system, a cellular telephone, a smart phone, a mobile device, a wearable device (e.g., a smart watch), a smart home device (e.g., a smart appliance), other smart devices, a web appliance, a network router, a network switch, a network bridge, or any machine capable of executing the instructions **1416**, sequentially or otherwise, that specify actions to be taken by the machine **1400**. Further, while only a single machine **1400** is illustrated, the term "machine" shall also be taken to include a collection of machines **1400** that individually or jointly execute the instructions **1416** to perform any one or more of the methodologies discussed herein.
[0074] The machine **1400** may include processors **1410**, memory/storage **1430**, and input/output (I/O) components **1450**, which may be configured to communicate with each other such as via a bus **1402**. In an example embodiment, the processors **1410** (e.g., a central processing unit (CPU), a reduced instruction set computing (RISC) processor, a complex instruction set computing (CISC)

processor, a graphics processing unit (GPU), a digital signal processor (DSP), an ASIC, a radio-frequency integrated circuit (RFIC), another processor, or any suitable combination thereof) may include, for example, a processor **1412** and a processor **1414** that may execute the instructions **1416**. The term "processor" is intended to include multi-core processors that may comprise two or more independent processors (sometimes referred to as "cores") that may execute instructions contemporaneously. Although FIG. **14** shows multiple processors, the machine **1400** may include a single processor with a single core, a single processor with multiple cores (e.g., a multi-core processor), multiple processors with a single core, multiple processors with multiples cores, or any combination thereof.

[0075] The memory/storage **1430** may include a memory **1432**, such as a main memory, or other memory storage, and a storage unit **1436**, both accessible to the processors **1410** such as via the bus **1402**. The storage unit **1436** and memory **1432** store the instructions **1416** embodying any one or more of the methodologies or functions described herein. The instructions **1416** may also reside, completely or partially, within the memory **1432**, within the storage unit **1436**, within at least one of the processors **1410** (e.g., within the processor's cache memory), or any suitable combination thereof, during execution thereof by the machine **1400**. Accordingly, the memory **1432**, the storage unit **1436**, and the memory of the processors **1410** are examples of machine-readable media.

[0076] As used herein, "machine-readable medium" means a device able to store instructions and data temporarily or permanently, and may include, but is not limited to, random-access memory (RAM), read-only memory (ROM), buffer memory, flash memory, optical media, magnetic media, cache memory, other types of storage (e.g., erasable programmable read-only memory (EEPROM)), and/or any suitable combination thereof. The term "machine-readable medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database, or associated caches and servers) able to store the instructions **1416**. The term "machine-readable medium" shall also be taken to include any medium, or combination of multiple media, that is capable of storing instructions (e.g., instructions **1416**) for execution by a machine (e.g., machine **1400**), such that the instructions, when executed by one or more processors of the machine (e.g., processors **1410**), cause the machine to perform any one or more of the methodologies described herein. Accordingly, a "machine-readable medium" refers to a single storage apparatus or device, as well as "cloud-based" storage systems or storage networks that include multiple storage apparatus or devices. The term "machine-readable medium" excludes signals per se.

[0077] Furthermore, the machine-readable medium is non-transitory in that it does not embody a propagating signal. However, labeling the tangible machine-readable medium "non-transitory" should not be construed to mean that the medium is incapable of movement; the medium should be considered as being transportable from one real-world location to another. Additionally, since the machine-readable medium is tangible, the medium may be considered to be a machine-readable device.

[0078] The I/O components **1450** may include a wide variety of components to receive input, provide output, produce output, transmit information, exchange information, capture measurements, and so on. The specific I/O components **1450** that are included in a particular machine will depend on the type of machine. For example, portable machines such as mobile phones will likely include a touch input device or other such input mechanisms, while a headless server machine will likely not include such a touch input device. It will be appreciated that the I/O components **1450** may include many other components that are not shown in FIG. **14**. The I/O components **1450** are grouped according to functionality merely for simplifying the following discussion and the grouping is in no way limiting. In various example embodiments, the I/O components **1450** may include output components **1452** and input components **1454**. The output components **1452** may include visual components (e.g., a display such as a plasma display panel (PDP), a light emitting diode (LED) display, a liquid crystal display (LCD), a projector, or a cathode ray tube (CRT)), acoustic components (e.g., speakers), haptic components (e.g., a vibratory

motor, resistance mechanisms), other signal generators, and so forth. The input components **1454** may include alphanumeric input components (e.g., a keyboard, a touch screen configured to receive alphanumeric input, a photo-optical keyboard, or other alphanumeric input components), point based input components (e.g., a mouse, a touchpad, a trackball, a joystick, a motion sensor, or another pointing instrument), tactile input components (e.g., a physical button, a touch screen that provides location and/or force of touches or touch gestures, or other tactile input components), audio input components (e.g., a microphone), and the like.

[0079] In further example embodiments, the I/O components **1450** may include biometric components **1456**, motion components **1458**, environmental components **1460**, or position components **1462** among a wide array of other components. For example, the biometric components **1456** may include components to detect expressions (e.g., hand expressions, facial expressions, vocal expressions, body gestures, or eye tracking), measure biosignals (e.g., blood pressure, heart rate, body temperature, perspiration, or brain waves), identify a person (e.g., voice identification, retinal identification, facial identification, fingerprint identification, or electroencephalogram based identification), and the like. The motion components **1458** may include acceleration sensor components (e.g., accelerometer), gravitation sensor components, rotation sensor components (e.g., gyroscope), and so forth. The environmental components **1460** may include, for example, illumination sensor components (e.g., photometer), temperature sensor components (e.g., one or more thermometers that detect ambient temperature), humidity sensor components, pressure sensor components (e.g., barometer), acoustic sensor components (e.g., one or more microphones that detect background noise), proximity sensor components (e.g., infrared sensors that detect nearby objects), gas sensors (e.g., gas detection sensors to detect concentrations of hazardous gases for safety or to measure pollutants in the atmosphere), or other components that may provide indications, measurements, or signals corresponding to a surrounding physical environment. The position components **1462** may include location sensor components (e.g., a Global Position System (GPS) receiver component), altitude sensor components (e.g., altimeters or barometers that detect air pressure from which altitude may be derived), orientation sensor components (e.g., magnetometers), and the like.

[0080] Communication may be implemented using a wide variety of technologies. The I/O components **1450** may include communication components **1464** operable to couple the machine **1400** to a network **1490** or devices **1470** via a coupling **1492** and a coupling **1472**, respectively. For example, the communication components **1464** may include a network interface component or other suitable device to interface with the network **1490**. In further examples, the communication components **1464** may include wired communication components, wireless communication components, cellular communication components, near field communication (NFC) components, Bluetooth® components (e.g., Bluetooth® Low Energy), Wi-Fix components, and other communication components to provide communication via other modalities. The devices **1470** may be another machine or any of a wide variety of peripheral devices (e.g., a peripheral device coupled via a Universal Serial Bus (USB)).

[0081] Moreover, the communication components **1464** may detect identifiers or include components operable to detect identifiers. For example, the communication components **1464** may include radio frequency identification (RFID) tag reader components, NFC smart tag detection components, optical reader components (e.g., an optical sensor to detect one-dimensional bar codes such as Universal Product Code (UPC) bar code, multi-dimensional bar codes such as Quick Response (QR) code, Aztec code, Data Matrix, Dataglyph, MaxiCode, PDF4140, Ultra Code, UCC RSS-2D bar code, and other optical codes), or acoustic detection components (e.g., microphones to identify tagged audio signals). In addition, a variety of information may be derived via the communication components **1464**, such as location via Internet Protocol (IP) geo-location, location via Wi-Fi® signal triangulation, location via detecting an NFC beacon signal that may indicate a particular location, and so forth.

Transmission Medium

[0082] In various example embodiments, one or more portions of the network **1490** may be an ad hoc network, an intranet, an extranet, a VPN, a LAN, a WLAN, a WAN, a WWAN, a MAN, the Internet, a portion of the Internet, a portion of the PSTN, a plain old telephone service (POTS) network, a cellular telephone network, a wireless network, a Wi-Fi® network, another type of network, or a combination of two or more such networks. For example, the network **1490** or a portion of the network **1490** may include a wireless or cellular network and the coupling **1492** may be a Code Division Multiple Access (CDMA) connection, a Global System for Mobile communications (GSM) connection, or another type of cellular or wireless coupling. In this example, the coupling **1482** may implement any of a variety of types of data transfer technology, such as Single Carrier Radio Transmission Technology (1xRTT), Evolution-Data Optimized (EVDO) technology, General Packet Radio Service (GPRS) technology, Enhanced Data rates for GSM Evolution (EDGE) technology, third Generation Partnership Project (3GPP) including 3G, fourth generation wireless (4G) networks, Universal Mobile Telecommunications System (UMTS), High Speed Packet Access (HSPA), Worldwide Interoperability for Microwave Access (WiMAX), Long Term Evolution (LTE) standard, others defined by various standard-setting organizations, other long range protocols, or other data transfer technology.

[0083] The instructions **1416** may be transmitted or received over the network **1490** using a transmission medium via a network interface device (e.g., a network interface component included in the communication components **1464**) and using any one of a number of well-known transfer protocols (e.g., HTTP). Similarly, the instructions **1416** may be transmitted or received using a transmission medium via the coupling **1472** (e.g., a peer-to-peer coupling) to the devices **1470**. The term “transmission medium” shall be taken to include any intangible medium that is capable of storing, encoding, or carrying the instructions **1416** for execution by the machine **1400**, and includes digital or analog communications signals or other intangible media to facilitate communication of such software.

[0084] Throughout this specification, plural instances may implement components, operations, or structures described as a single instance. Although individual operations of one or more methods are illustrated and described as separate operations, one or more of the individual operations may be performed concurrently, and nothing requires that the operations be performed in the order illustrated. Structures and functionality presented as separate components in example configurations may be implemented as a combined structure or component. Similarly, structures and functionality presented as a single component may be implemented as separate components. These and other variations, modifications, additions, and improvements fall within the scope of the subject matter herein.

[0085] Although an overview of the inventive subject matter has been described with reference to specific example embodiments, various modifications and changes may be made to these embodiments without departing from the broader scope of embodiments of the present disclosure. Such embodiments of the inventive subject matter may be referred to herein, individually or collectively, by the term “invention” merely for convenience and without intending to voluntarily limit the scope of this application to any single disclosure or inventive concept if more than one is, in fact, disclosed.

[0086] The embodiments illustrated herein are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed. Other embodiments may be used and derived therefrom, such that structural and logical substitutions and changes may be made without departing from the scope of this disclosure. The Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by the appended claims, along with the full range of equivalents to which such claims are entitled.

[0087] As used herein, the term “or” may be construed in either an inclusive or exclusive sense. Moreover, plural instances may be provided for resources, operations, or structures described

herein as a single instance. Additionally, boundaries between various resources, operations, modules, engines, and data stores are somewhat arbitrary, and particular operations are illustrated in a context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within a scope of various embodiments of the present disclosure. In general, structures and functionality presented as separate resources in the example configurations may be implemented as a combined structure or resource. Similarly, structures and functionality presented as a single resource may be implemented as separate resources. These and other variations, modifications, additions, and improvements fall within a scope of embodiments of the present disclosure as represented by the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

[0088] In this document, the terms “a” or “an” are used, as is common in patent documents, to include one or more than one, independent of any other instances or usages of “at least one” or “one or more.” In the appended claims, the terms “including” and “in which” are used as the plain-English equivalents of the respective terms “comprising” and “wherein.” Also, in the following claims, the terms “including” and “comprising” are open-ended; that is, a system, device, article, or process that includes elements in addition to those listed after such a term in a claim are still deemed to fall within the scope of that claim. Moreover, in the following claims, the terms “first,” “second,” “third,” and so forth are used merely as labels, and are not intended to impose numerical requirements on their objects.

Claims

1-20. (canceled)

21. A method comprising: accessing first application data of a first application, the first application having one or more data validation rules; accessing an application data schema comprising a set of constraints on second application data consumed by a second application; determining the one or more data validation rules do not include at least one constraint from the set of constraints in the application data schema; and in response to determining the one or more data validation rules do not include at least one constraint from the set of constraints in the application data schema, preventing synchronization of the second application data with the first application data; wherein the method is performed by one or more processors.

22. The method of claim 21, further comprising: determining that an additional validation rule has been associated with the first application; and in response to the determination, synchronizing the second application data of the second application with the first application data of the first application.

23. The method of claim 22, wherein the first application data includes data of a plurality of cells, wherein the method further comprises: causing display of a window that includes a selectable list of predefined validation rules for association with one or more cells in the plurality of cells; and receiving a user selection, from the user selectable list, of a predefined validation rules for association with the one or more cells in the plurality of cells, the predefined validation rule corresponding to the additional validation rule.

24. The method of claim 22, further comprising: receiving a modification to at least one data entry in the first application data; validating the at least one data entry according to the one or more data validation rules and the additional validation rule; and synchronizing the second application data consumed by the second application with the first application data such that the second application data includes the modification to the at least one data entry in the first application data.

25. The method of claim 22, further comprising: receiving, from a client device, user specified logic, the user specified logic comprising a name, a validator, and a value type; and receiving, from the client device, one or more user specified configuration parameters; wherein the user specified logic and the user specified configuration parameters correspond to the additional validation rule.

26. The method of claim 25, wherein the validator includes a validator associated with the value type; wherein the validator includes at least one selected from a group consisting of a date validator and a geo-coordinate validator.
27. The method of claim 21, further comprising causing display of a synchronization interface to present a status of synchronization of the first application data with the second application data.
28. The method of claim 21, wherein: the application data schema is a first application data schema; the set of constraints is a first set of constraints; and the method further comprises: accessing a second application data schema comprising a second set of constraints on third application data consumed by a third application; determining whether the one or more data validation rules include the second set of constraints; and in response to determining that the one or more data validation rules include the second set of constraints, synchronizing the third application data consumed by the third application with the first application data.
29. The method of claim 28, further comprising: in response to determining the one or more data validation rules do not include at least one constraint from the second set of constraints in the second application data schema, preventing synchronization of the third application data with the first application data;
30. The method of claim 21, further comprising: receiving, from a client device running the first application, a modification to at least one data entry in the first application data; validating the at least one data entry according to the one or more data validation rules; and synchronizing the second application data consumed by the second application with the first application data such that the second application data includes the modification to the at least one data entry in the first application data.
31. The method of claim 21, wherein the first application data including data of a plurality of cells, wherein the method further comprises: validating the first application data according to the one or more data validation rules; determining at least one cell of the plurality of cells includes an invalid data entry; and in response to determining the at least one cell of the plurality of cells includes the invalid data entry, causing the at least one cell to be visually distinguished from remaining cells during display of the first application data.
32. A system comprising: one or more processors of a machine; and one or more memories storing instructions that, when executed by at least one processor among the one or more processors, cause the system to perform operations comprising: accessing first application data of a first application, the first application having one or more data validation rules; accessing an application data schema comprising a set of constraints on second application data consumed by a second application; determining the one or more data validation rules do not include at least one constraint from the set of constraints in the application data schema; and in response to determining the one or more data validation rules do not include at least one constraint from the set of constraints in the application data schema, preventing synchronization of the second application data with the first application data.
33. The system of claim 32, wherein the operations further comprise: determining that an additional validation rule has been associated with the first application; and in response to the determination, synchronizing the second application data of the second application with the first application data of the first application.
34. The system of claim 33, wherein the first application data includes data of a plurality of cells, wherein the operations further comprise: causing display of a window that includes a selectable list of predefined validation rules for association with one or more cells in the plurality of cells; and receiving a user selection, from the user selectable list, of a predefined validation rules for association with the one or more cells in the plurality of cells, the predefined validation rule corresponding to the additional validation rule.
35. The system of claim 33, wherein the operations further comprise: receiving a modification to at least one data entry in the first application data; validating the at least one data entry according to

the one or more data validation rules and the additional validation rule; and synchronizing the second application data consumed by the second application with the first application data such that the second application data includes the modification to the at least one data entry in the first application data.

36. The system of claim 33, wherein the operations further comprise: receiving, from a client device, user specified logic, the logic comprising a name, a validator, and a value type; and receiving, from the client device, one or more user specified configuration parameters; wherein the user specified logic and the user specified configuration parameters correspond to the additional validation rule.

37. The system of claim 36, wherein the validator includes a validator associated with the value type; wherein the validator includes at least one selected from a group consisting of a date validator and a geo-coordinate validator.

38. The method of claim 32, wherein the operations further comprise causing display of a synchronization interface to present a status of synchronization of the first application data with the second application data.

39. The system of claim 32, wherein: the application data schema is a first application data schema; the set of constraints is a first set of constraints; and the operations further comprise: accessing a second application data schema comprising a second set of constraints on third application data consumed by a third application; determining whether the one or more data validation rules include the second set of constraints; and in response to determining that the one or more data validation rules include the second set of constraints, synchronizing the third application data consumed by the third application with the first application data.

40. A non-transitory machine-readable storage medium embodying instructions that, when executed by at least one processor of a machine, cause the machine to perform operations comprising: accessing first application data of a first application, the first application having one or more data validation rules; accessing an application data schema comprising a set of constraints on second application data consumed by a second application; determining the one or more data validation rules do not include at least one constraint from the set of constraints in the application data schema; and in response to determining the one or more data validation rules do not include at least one constraint from the set of constraints in the application data schema, preventing synchronization of the second application data with the first application data.