



(19) **United States**

(12) **Patent Application Publication**
Abileah et al.

(10) **Pub. No.: US 2014/0100956 A1**

(43) **Pub. Date: Apr. 10, 2014**

(54) **PROVIDING FAST ALERTS FOR RAPIDLY CHANGING REAL ESTATE DATA**

(52) **U.S. Cl.**
CPC *G06Q 50/16* (2013.01); *G06Q 30/0257* (2013.01)

(71) Applicant: **REDFIN Corporation**, Seattle, WA (US)

USPC **705/14.55**

(72) Inventors: **Shahaf Abileah**, Seattle, WA (US);
April Alexander, Seattle, WA (US);
Vincent Chang, Seattle, WA (US);
Michael Smedberg, Seattle, WA (US)

(57) **ABSTRACT**

(21) Appl. No.: **13/907,724**

(22) Filed: **May 31, 2013**

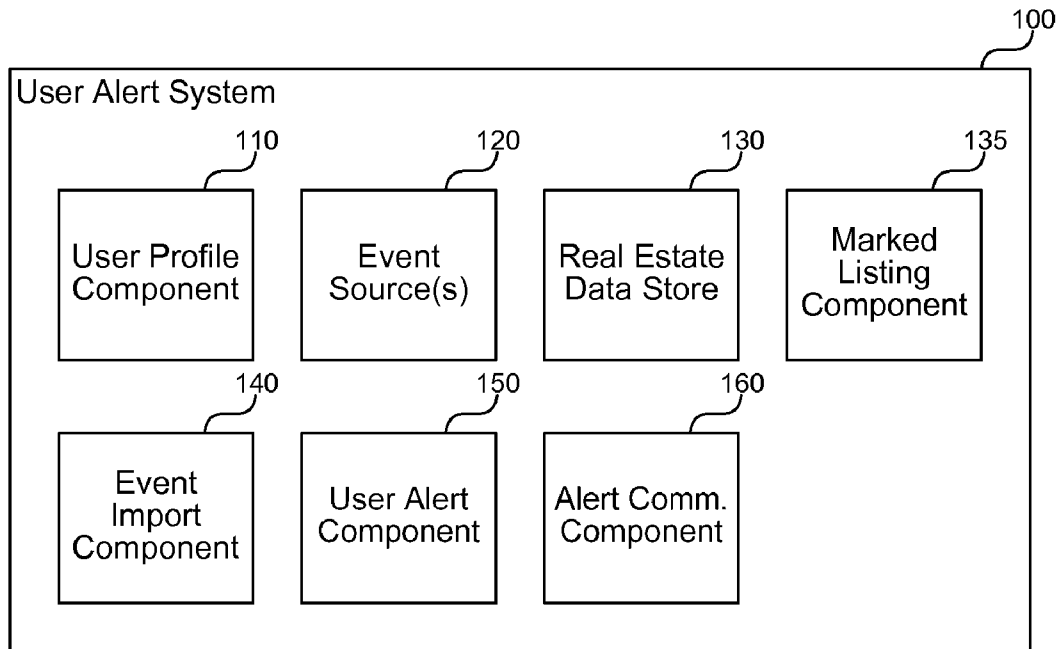
Related U.S. Application Data

(60) Provisional application No. 61/710,606, filed on Oct. 5, 2012.

Publication Classification

(51) **Int. Cl.**
G06Q 50/16 (2006.01)
G06Q 30/02 (2006.01)

A user alert system is described herein that provides direct alerts to users of a real estate website each time a single event meets one of a user's criteria. Unlike periodic summary reports that summarize all events within a regular time period, direct or "instant" alerts are delivered as soon as possible after a condition is met for a subject property or other source of the event, so that the user receives the most timely notification possible. Users can create a set of search criteria and then save the search for future reference. The user alert system leverages these search criteria to identify any item that matches the user's search criteria and provide a real-time notification in a timely manner upon any real estate event occurring that matches the user's search criteria. Thus, the user alert system allows users to find out about changes much more rapidly.



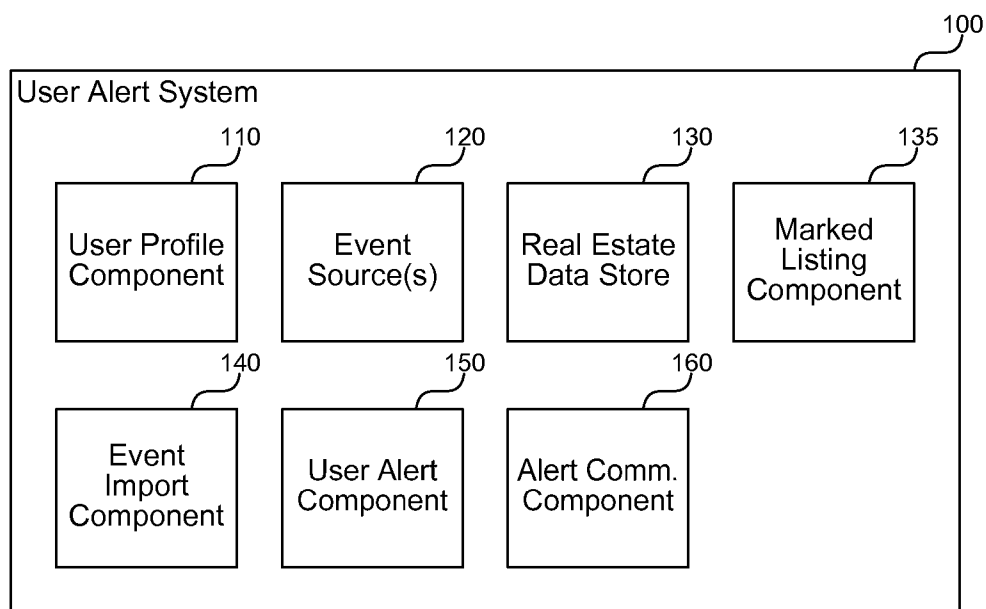


FIG. 1

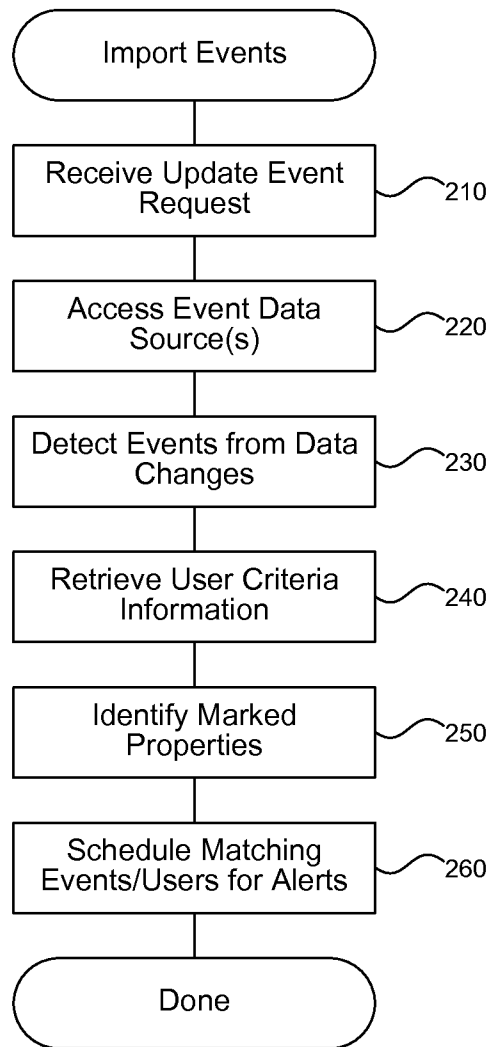


FIG. 2

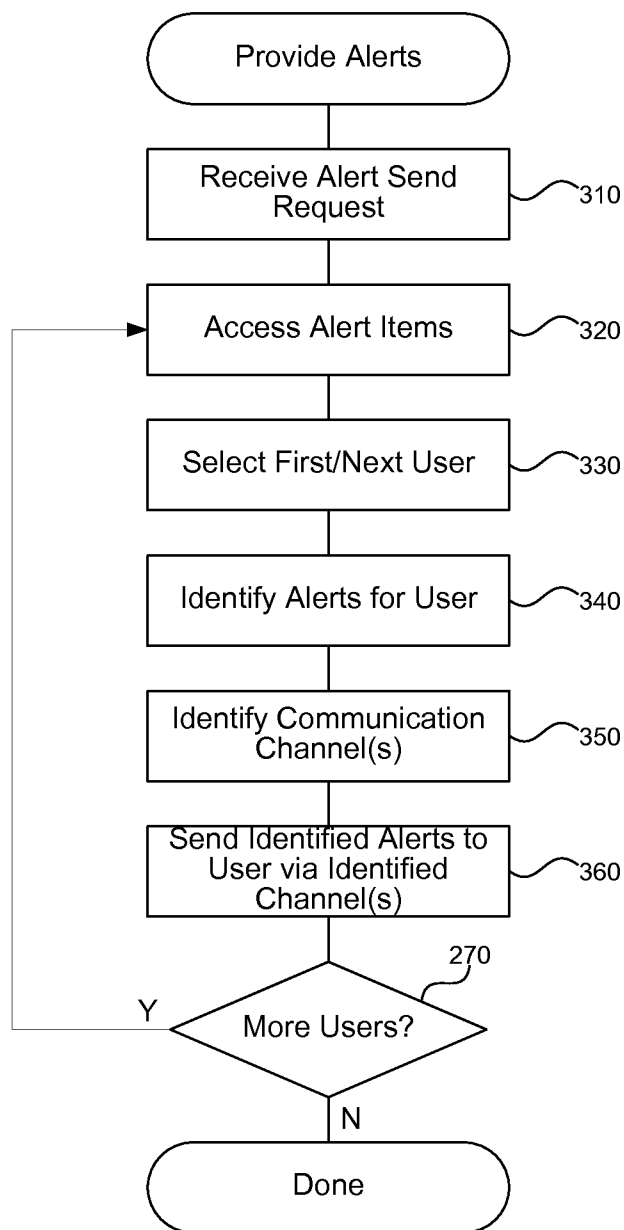


FIG. 3

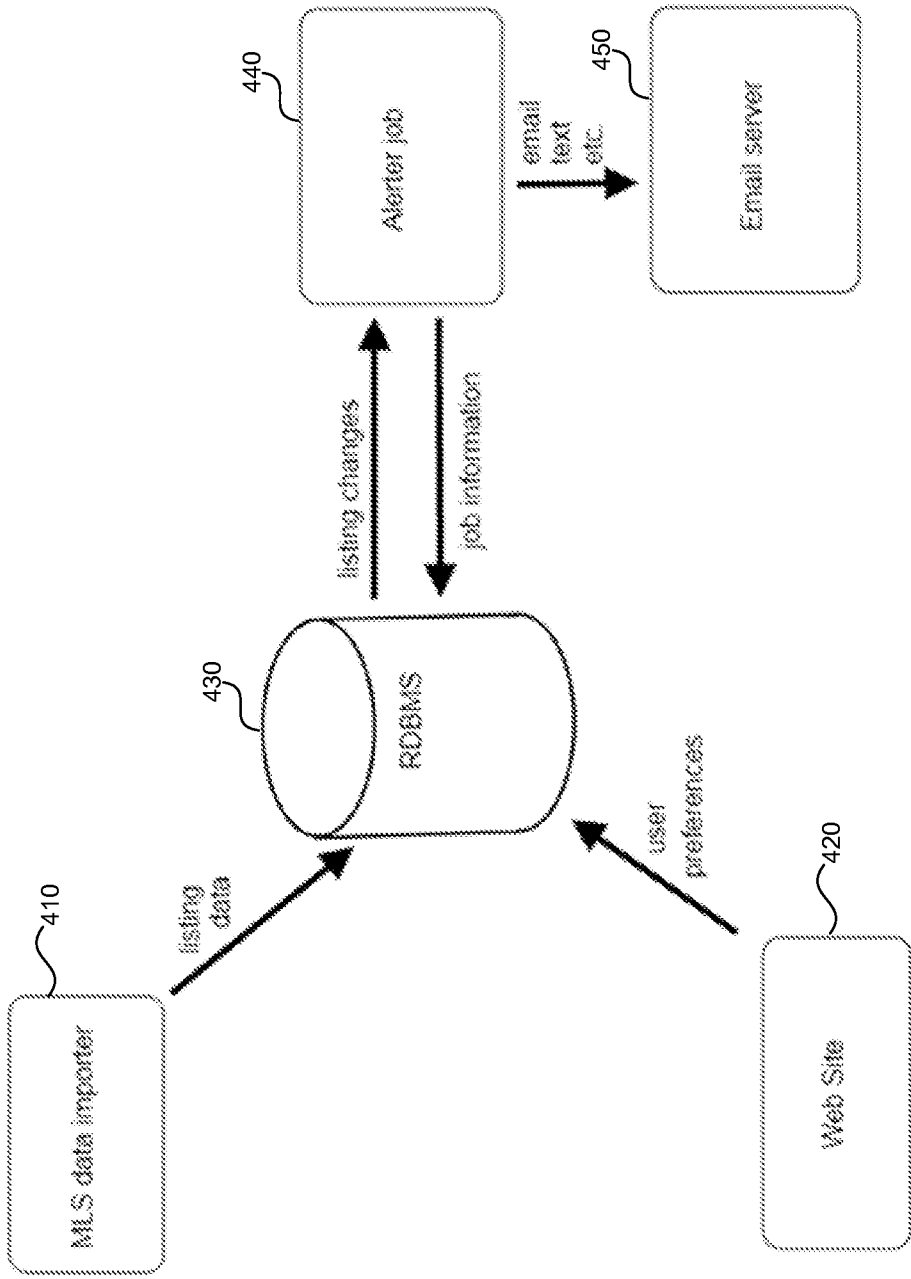


FIG. 4

PROVIDING FAST ALERTS FOR RAPIDLY CHANGING REAL ESTATE DATA

PRIORITY CLAIM

[0001] The present application claims priority from U.S. Provisional Application No. 61/710,606, filed Oct. 5, 2012, which is incorporated by reference as if fully set forth herein.

BACKGROUND OF THE INVENTION

[0002] Real estate transactions often involve a buyer, a seller, one or more agents (e.g., a listing agent for the seller and buyer's agent for the buyer), and numerous service providers (e.g., escrow professionals, appraisers, home inspectors, lenders, and so forth). In a typical purchase transaction, the seller lists a property using a listing server, often through a listing agent, while buyers go through a search process to identify potential properties to buy. Once the buyer identifies a property the buyer wants to buy, the buyer makes an offer to the seller, potentially each through their respective agents. The seller may either accept the offer, counteroffer at a different price or other terms, or reject the offer. Once an offer or counteroffer has been accepted, the transaction is typically considered pending and on the way to closing pending inspections, resolving various contingencies, securing funds from a lender, and so forth.

[0003] In some places and times, the market for residential real estate is fast-moving and, consequently, difficult to keep up with in terms of current and relevant information. Some homes sell within hours of going on the market, or very soon after price reductions and other changes. Purchasers (and others interested in the real estate market) are helped by very rapid access to information to negotiate the best deals. However, they may not have time to constantly review sources of real estate information (e.g., web sites, publications, and so on).

[0004] Some real estate web sites and other applications allow users to create a set of search criteria (e.g., limited to a particular region, price band, or other criteria), and then save the search. The user can then re-run that search at any time. These sites also often run a periodic process to publish reports based on these searches (e.g., nightly, or weekly). Users can subscribe to receive email alerts on aforementioned saved searches that contain information about listings of interest (including new listings, updated listings, price changes, sales, and the like). In addition, users can mark particular listings as "favorite" listings. When alerts are delivered, users will be notified of changes to listings that match a saved search or are a favorite. However, these reports are primarily oriented on summarizing a variety of information at a regular time, and even nightly reports may be too infrequent in a fast-changing real estate environment.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. 1 is a block diagram that illustrates components of the user alert system, in one embodiment.

[0006] FIG. 2 is a flow diagram that illustrates processing of the user alert system to import events from one or more data sources of interest to one or more users, in one embodiment.

[0007] FIG. 3 is a flow diagram that illustrates processing of the user alert system to provide alerts to one or more users of a real estate data system, in one embodiment.

[0008] FIG. 4 is a data flow diagram that illustrates components of the user alert system and how data flows between or among them, in one embodiment

DETAILED DESCRIPTION

[0009] Embodiments of the invention may be operational with numerous general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, micro-processor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[0010] Embodiments of the invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer and/or by computer-readable media on which such instructions or modules can be stored. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

[0011] Embodiments of the invention may include or be implemented in a variety of computer readable media. Computer readable media can be any available media that can be accessed by a computer and includes both volatile and non-volatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer readable media.

[0012] According to one or more embodiments, the combination of software or computer-executable instructions with a computer-readable medium results in the creation of a machine or apparatus. Similarly, the execution of software or

computer-executable instructions by a processing device results in the creation of a machine or apparatus, which may be distinguishable from the processing device, itself, according to an embodiment.

[0013] Correspondingly, it is to be understood that a computer-readable medium is transformed by storing software or computer-executable instructions thereon. Likewise, a processing device is transformed in the course of executing software or computer-executable instructions. Additionally, it is to be understood that a first set of data input to a processing device during, or otherwise in association with, the execution of software or computer-executable instructions by the processing device is transformed into a second set of data as a consequence of such execution. This second data set may subsequently be stored, displayed, or otherwise communicated. Such transformation, alluded to in each of the above examples, may be a consequence of, or otherwise involve, the physical alteration of portions of a computer-readable medium. Such transformation, alluded to in each of the above examples, may also be a consequence of, or otherwise involve, the physical alteration of, for example, the states of registers and/or counters associated with a processing device during execution of software or computer-executable instructions by the processing device.

[0014] As used herein, a process that is performed “automatically” may mean that the process is performed as a result of machine-executed instructions and does not, other than the establishment of user preferences, require manual effort.

[0015] A user alert system is described herein that provides direct alerts to users of a real estate website or application each time a single event meets one of a user’s criteria. Unlike periodic summary reports that summarize all events within a regular time period, direct or “instant” alerts are delivered as soon as possible after a condition is met for a subject property or other source of the event, so that the user receives the most timely notification possible. Users can create a set of search criteria (e.g., limited to a particular region, price band, house type, and so forth), and then save the search for future reference. The user alert system leverages these search criteria to identify any item that matches the user’s search criteria and provide a real-time notification in a timely manner upon any real estate event occurring that matches the user’s search criteria. Although in some implementations the notifications may still be periodic for efficient implementation on the backend, the period is far more frequent than that available today (e.g., every 15 minutes versus once a day). Notifications may be provided via a variety of notification channels, such as an email message, a text message, a push notification on a mobile device, a notification center of a desktop operating system, and so on.

[0016] In some embodiments, listing information is extracted from a multiple listing service (MLS) data provider, such as a regional MLS database. Users are notified of relevant changes in the MLS very soon after the changes are made in the MLS. In some embodiments, users may choose which information they are interested in receiving (e.g., they may set search parameters—“listings under \$200 k”—and they may restrict alerts to selected subject matter—e.g., “only alert me on new listings, not on price changes”). Information from multiple MLSs may also be aggregated into periodic alerts. In some embodiments, the system applies an adaptive quality to information, either through default settings or user-preference settings, so that information that gets stale quickly is sent very frequently, while information for which timeli-

ness is less relevant is sent less frequently (e.g., new listings every 15 minutes, price changes every 15 minutes, status changes every 15 minutes, open house changes daily, reviews by real estate agents every 15 minutes for favorite listings and daily for others). In some embodiments, users may mark particular listings as not interesting by crossing them out in a user interface to mark them as an “anti-favorite”. Users are then not notified of changes to anti-favorite listings, even if the listing matches a saved search. Users may choose listings of interest via a search (e.g., “listings in Oakland under \$200 k”), by choosing particular listings (e.g., indicating that a listing is a “favorite”), or by other methods provided by the real estate application. Thus, the user alert system allows users to find out about changes much more rapidly, which may mean the difference between a user making the first offer on a listing or not and ultimately finding the house the user wants.

[0017] The following paragraphs provides details of an example implementation of the user alert system, though those of ordinary skill in the art will recognize that any particular implementation may depart from this example while staying within the scope of those principles of operation of the system described herein.

[0018] In the example embodiment, information from MLSs is periodically imported into a relational database. One or more “importer” jobs run as frequently as reasonable based on a tradeoff between scalability of the backend (as well as any limitations of the MLS provider) and getting alerts to users as frequently as possible. For MLSs that offer real-time information, the importer jobs may run very frequently (e.g., every 5 minutes). When new information is found, the importer job records the change to a table in a local database of the real estate application. The data is recorded as a delta—a record of what changed and when it changed (as opposed to simply recording the current value).

[0019] An “alerts” job also runs periodically (e.g., every 5 or 15 minutes). Each time the alerts job runs, the job calculates the period for which the job is reporting. This period typically starts at the end of the period of the previous job, and ends at the current time. The alerts job finds all of the listing changes that happened during the relevant period. The job correlates the identified changes with the users who are interested in those listings (either because the listing is a favorite, or because the listing matches a saved search). In some embodiments, the alerts job may be implemented as a structured query language (SQL) process that runs periodically in a relational database.

[0020] The job iterates through the users for which at least one listing was identified (although described serially for ease of explanation, those skilled in the art will recognize that a similar parallel process could be implemented to handle batches or individual users at the same time). For each user, the job creates a description (e.g., a text string) that describes all of the relevant listing changes for listings of interest to the user. The description may also include marketing messages (e.g., advertisements), or other information. For each user, the system sends an alert to the user of the relevant information. Notification may occur through email or any other electronic medium—short message service (SMS) messages, Twitter messages, push notifications, and so forth. The format of the notification may differ by medium (e.g., a long hypertext markup language (HTML) document for email and a short, text-only message for SMS).

[0021] FIG. 1 is a block diagram that illustrates components of the user alert system, in one embodiment. The system **100** includes a user profile component **110**, one or more event sources **120**, a real estate data store **130**, a marked listing component **135**, an event import component **140**, a user alert component **150**, and an alert communication component **160**. Each of these components is described in further detail herein.

[0022] The user profile component **110** stores user profile information for one or more users of the system **100**. The profile information may include information such as authentication information (e.g., a user identifier and password), contact information (e.g., an email address, phone number, postal address, and so forth), preference information (e.g., what areas a buyer is looking in for purchases, types of properties of interest, and so forth), transactions the user is a party to, saved searches, favorite listings, and so forth. Users may also have a type tracked by the system, such as buyer, seller, service provider, agent, and the like. Any user may operate in several categories at the same time, and may be associated with notifications related to each. For example, a person may be selling a house in one area while buying a house in another. In some cases, the system may create separate alerts or classifications of events in an alert that allow a user to be separately notified of events related to only one particular scope at some times and all events at others, upon request.

[0023] The one or more event sources **120** include various sources of real estate related information. These may include data streams from a multiple listing service (MLS), a county information data stream (e.g., sales data, assessor information, tax payments, and the like), transaction information, broker/agent information, and so forth. The real estate website may use these sources of data for everyday operation of the website, such as to provide real estate searches, property information, value estimates, or other purposes. The user alert system described herein uses these event sources to identify and quickly notify the user of events of interest for each user of the real estate website.

[0024] The real estate data store **130** is a data storage facility that stores information from the event sources, user profile information, and user notification information. The data storage facility may include one or more files, file systems, hard drives, databases, cloud based storage services, or other facilities for storing data persistently over time. For example, the real estate data store **130** may be implemented as a relational database with various tables related to users, properties, transactions, events, and so forth. The system **100** may use queries or other data access methods to periodically identify information stored in the data store **130** that may be of interest to each user, and to provide notifications to each user. In some cases, for efficiency the system may determine groups of users with similar characteristics (e.g., each looking for a similar home in the same area), and perform a unified query to identify events for the group. Although some actions for multiple users may be performed together, the system **100** provides a high level of personalization such that each user may have a different set of events for which the user is notified.

[0025] The marked listing component **135** manages one or more listings marked by users as favorites or non-favorites. While browsing properties, the system **100** may allow users to mark listings of interest as favorites, as well as marking listings not of interest as non-favorites. Thus, any listing in the system **100** may at any time for a particular user by unmarked

(e.g., perhaps new or simply not marked by the user), a favorite, or an anti-favorite. The marked listing component **135** stores markings of properties associated with each user in the real estate data store **130**. The system **100** uses this information to provide alerts for properties marked as favorites, and to avoid providing alerts for those properties marked as anti-favorites. Those properties that are unmarked may generate alerts upon meeting other criteria, such as qualifying as a search result to a saved search.

[0026] The event import component **140** detects events from the event sources **120** and schedules the detected events for notification to applicable users. In some embodiments, the component **140** may store pending notifications in a data structure of the real estate data store **130**. The event import component **140** may detect events from an MLS, such as price increases, changes in status of a listing (e.g., from “for sale” to “under contract” to “pending” to “sold”), new photos available, changes in agent remarks about a listing, notes from a buyer or seller’s own agent to the buyer or seller, and so forth. The component **140** may also detect events from other sources, such as a change in assessment or tax value from a county data stream, a rise in the level of interest indicated by page views on a real estate site, new transaction information available from a broker’s transaction data stream, or any other event deemed relevant by an implementer of the system **100**. The component **140** then schedules these events for notification of users, either with complete information or with a sufficient amount of reference information to retrieve further information about the event when the notification occurs. The component **140** may store type or source information that indicates where the event came from and what type of event it is. This information is useful later for identifying and culling events relevant to a particular user’s notifications.

[0027] The user alert component **150** periodically analyzes the real estate data store and detected events to identify events relevant to particular users and to communicate those events to the particular users. For example if a particular user is interested in new listings in a geographic area, then the component **150** associates any new listings detected in that area with the user. The user may be a buyer, seller, agent, service provider, or other user of the system **100**. The component **150** may work in conjunction with configuration information stored in a user’s profile that specifies types of events for which a particular user is interested in receiving notifications. For example, a service provider or agent may receive a large number of events, and may want to configure his or her profile so that not all events generate notifications, while a buyer or seller that is actually seeking a sale may want to receive every event relevant to that user. The user-alert component **150** may search (e.g., via a query) various sub-stores of the data store **130**, such as tables for listings, transactions, and so forth, to identify events relevant to particular users. The component **150** may regularly run a process for each user that identifies events of relevance to that user, or may group users where the data to be identified for each user is similar (e.g., same types of listings in a given area). The result is a set of events associated with each user that the user is rapidly notified of without the user manually seeking out the information.

[0028] The alert communication component **160** handles communication of identified alerts to each user. The alert communication may use user profile information to determine a format and channel of alert to use for a particular user. For example, some users may request notification by email, while others may prefer a text message. Some users may

prefer alerts by all of the available alert methods, so that the user is more likely to see the alert in a timely manner. The alert communication component **160** may include knowledge of various protocols and services for sending alerts, such as simple message transport protocol (SMTP) for email, SMS or multimedia message service (MMS) for text messages, push notifications for various mobile or other computing platforms, and so forth. The alert communication component **160** may also provide additional services, such as resending dropped alerts, receiving read receipts that indicate which alerts were read, and so forth.

[0029] FIG. 2 is a flow diagram that illustrates processing of the user alert system to import events from one or more data sources of interest to one or more users, in one embodiment. Beginning in block **210**, the system receives a request to check for new events for which users may have requested to be notified. For example, the system may run a process on a tight frequency (e.g., every five minutes) that requests new events to be imported into the system. The request may identify one or more data sources for which new information is rapidly available, such as an MLS. In response to the request, the system performs the following steps to discover new event information and prepare the information for notification to users.

[0030] Continuing in block **220**, the system accesses one or more event data sources to retrieve information that, if changed, is relevant for an alert that at least one user has requested to receive. The system connects to each event source. The event source may provide a website of its own, a web service application-programming interface (API), a file transfer protocol (FTP) site, a remote database connection, or data sharing protocol to which the system can connect to receive new data. The system connects according to the protocol of the particular source, and retrieves new data from that source. The system may do this for each of a series of sources to update a local data store of aggregate real estate information (e.g., information from multiple sources about a set of properties).

[0031] The system may receive information in one format, such as bulk information about each of the tax parcels in a county, and then may separate this information into individual events (e.g., property A's tax assessed value increased). The system may then store these events in an events table or other data structure of a data store, such as a relational database. The system may also store the bulk data and later extract events by querying one or more records associated with the bulk data. For example, the system may compare two sets of bulk data from different periods to form a "diff" view of items that differ, and then turn these differences into events. In some cases, a failure to change may also be an event, such as an unsold property with a listing price that has not dropped in more than 60 days. The system can create events for any data occurrence of interest to a particular implementation of the system.

[0032] Continuing in block **230**, the system compares retrieved information to previously retrieved information to identify one or more data changes and generates an event for each data change. The system may store the results of retrieving information each time information is retrieved, and may keep a history of such information, so that comparisons can be performed on various bases (e.g., hourly, daily, weekly, monthly, and yearly). Changes may include changes in the status of a listing, changes in price, a new listing entirely, changes in terms of sale, and so forth. An event typically

indicates a type of change (e.g., what changed), and a value associated with the change (e.g., how did it change). This information becomes the basis for a description of the event (e.g., "listing price decreased \$5,000").

[0033] Continuing in block **240**, the system accesses user criteria information that specifies one or more criteria for at least one user for which the user has requested to be notified upon the identification of a matching property data change. The system may access one or more stored user profiles that stores data for each user. A user profile may include saved property searches, a list of favorite properties, a list of anti-favorite properties, general criteria of properties the user likes, and so on. The system accesses this information and compares it to the identified one or more data changes. If a property has an associated data change and the property is one that matches a user's favorites or other criteria, then the user may receive an alert from the system describing the change.

[0034] Continuing in block **250**, the system identifies one or more properties marked by at least one user for which changes to the properties are requested by the user to generate an alert. Marked properties are those that the user has previously identified as favorites and for which the user always wants to be notified of changes. Marked properties may also include those identified as anti-favorites for which the user never wants to be notified of changes. The system considers the user's marked properties to ensure that the user receives notifications for those properties that the user is most interested in (and does not receive those in which the user has no interest).

[0035] Continuing in block **260**, the system schedules alerts to notify users of generated events that match at least one accessed user criteria or marked property. The system may store a list of pending alerts, such as in a database table, or may send the alerts immediately upon determining that an alert should be sent. The system may separate the process of identifying alerts from the process of sending the alerts. This allows the system to quickly identify alerts to be sent without being delayed by any errors or other difficulties that can occur during sending. FIG. 3 describes a detailed process for providing alerts that have been identified and scheduled to users of the system. After block **260**, these steps conclude.

[0036] FIG. 3 is a flow diagram that illustrates processing of the user alert system to provide alerts to one or more users of a real estate data system, in one embodiment. Beginning in block **310**, the system receives a request to send pending alerts to users of the real estate data system. For example, the system may initiate the process of sending alerts immediately after the process of identifying alerts concludes (see, e.g., FIG. 2). Because the system is typically designed to provide rapid alerts to users, the system does not introduce unnecessary delay in the process of sending alerts, but may subdivide the work of sending alerts in a manner that is more efficient for processing on a backend data server.

[0037] Continuing in block **320**, the system accesses one or more pending alert items previously detected based on one or more changes in data provided to the real estate data system by a real estate data provider. The alert items may have a general scope not yet associated with any particular user. For example, each listing that has changed may occupy a row in an events database table that indicates what changed about the listing (e.g., listing price decreased, open house scheduled, status changed to pending, and the like). In some embodiments, the system provides adaptive alerts to users such that only those alerts that are deemed to be of a time sensitive

nature (e.g., price changes, status changes, new listings) are provided in real time while other events that are less time sensitive (e.g., open house scheduled, tax assessment changed) are provided in a more traditional, longer period (e.g., daily or weekly).

[0038] Continuing in block **330**, the system selects a first user to which to provide alerts. On subsequent iterations, the system selects the next user of the system that has requested alerts. Although shown serially for ease of explanation, those of ordinary skill in the art will recognize that the system could notify users in parallel, such as by notifying all users at once or handling blocks of users at a time. Such implementations typically strike a balance between faster alerts to users and managing resources on the backend effectively. Upon selecting the user, the system accesses stored profile information associated with the user, such as saved property searches and properties marked for inclusion and/or exclusion from notifications.

[0039] Continuing in block **340**, the system identifies one or more pending alerts applicable to the selected user. An alert may be deemed applicable based on matching a saved real estate search of the user, pertaining to a favorite property marked by the user, or based on other criteria. A data change related to a property that otherwise satisfies the user's criteria may also be ignored and alerts not sent to the user based on the user having previously marked the property as one to ignore (i.e., an anti-favorite). Events of interest to a user may include any listing changes (or lack of changes) for listings that satisfy a real estate search query saved by the user in the user's profile, events related to a transaction to which the user is a party, events based on behavior of other users (e.g., properties going under contract have risen in an area), and so forth. The system identifies events by matching all of the available event data streams to preference and other configuration information known about the user.

[0040] Continuing in block **350**, the system identifies one or more communication channels requested by the user for receiving the pending alerts. A user's profile includes contact information as well as contact preference information. For example, a user may specify an email address, mobile number for receiving text messages, device identifier for receiving push notifications, and so forth. The system may provide varying levels of configurability. In some embodiments, the system may allow a user to map each alert type to a communication channel, so that the user, for example, receives new listings via text message and email, but changes in existing listings only by email. The user can then specify how he or she wants to be notified of each type of alert. A user may also have particular periods during which the user wants a higher level of alerts (e.g., daytime business hours) and other periods where less important alerts are held or not sent (e.g., during sleeping hours). In a hot real estate market, a user may opt for a particularly high level of alerts, while in others the user may be satisfied with a more moderate level of alerts.

[0041] Continuing in block **360**, the system sends the identified alerts to the selected user via the identified communication channels, wherein the system alerts users rapidly without waiting for the expiration of a specified period or the occurrence of multiple events before sending information to the user. The system may also mark alerts has having been sent or clear them from a list of pending alerts upon sending them to the user, so that the user does not receive repeat alerts for the same event. Past real estate systems wait for a period to expire (e.g., nightly) before even checking for alerts for

users. The user alert system described herein checks for changed data frequently, and then works to get alerts to users quickly thereafter. The goal of other systems is to batch all of the changes that have occurred during the period in a digest, so that the user is only bothered once per day or other period. In contrast, the system herein does not wait to batch events, and while multiple events may naturally occur in the same notification cycle, the system herein is intentionally noisier and may send many alerts per day that include on a single data change each.

[0042] Continuing in decision block **370**, if there are more users then the system loops to block **330** to select the next users for which to identify alerts. When all of the users have been sent any pending alerts, the system completes to wait for the next round when the process begins again. After block **370**, these steps conclude.

[0043] FIG. 4 is a data flow diagram that illustrates components of the user alert system and how data flows between them, in one embodiment. An MLS data importer **410** retrieves data to detect changes from an MLS on a fast period (e.g., multiple times per hour). The MLS data importer **410** stores imported data in a relational database **430** associated with a real estate website **420**. The real estate website **420** provides one or more user interfaces for receiving user preferences, such as saved real estate search criteria, identified favorite properties, and the like. The system runs an alerter job **440** to identify listing changes in the relational database **430** and send alerts to users and/or parties designated by such users, such as via email server **450**. The system runs the MLS data importer **410** and alerter job **440** processes continuously to detect frequent changes to listing data that may occur throughout the day, and to ensure that users of the real estate website **420** are the first to know about new and/or changed listings.

[0044] From the foregoing, it will be appreciated that specific embodiments of the user alert system have been described herein for purposes of illustration, but that various modifications may be made without deviating from the spirit and scope of the invention. For example, a user who is a potential property purchaser may, in an embodiment, be given the option of selecting his/her agent to likewise or solely receive alerts/notifications as a means of enabling faster agent response (e.g., making a purchase offer on a property) on the user's behalf without requiring the user to consider the alert. Additionally, such user who is a potential property purchaser may, in an embodiment, be given the option of configuring one or more components of system **100** described herein to automatically generate and send a purchase offer message (e.g., email, text, etc.) to an agent for the seller of the property in response to receiving or the generation of an alert/notification, as described above herein, concerning the property. Such an embodiment enables faster response (e.g., making a purchase offer on a property) on behalf of a user, without requiring the user to consider the alert, in response to threshold criteria being met concerning the property. Accordingly, the invention is not limited except as by the appended claims.

1. A computer-implemented method to import real estate events from one or more data sources of interest to one or more users of a real estate application, the method comprising:

receiving a request to check for new events for which users may have requested to be notified;

accessing one or more event data sources to retrieve information that, if changed, is relevant for an alert that at least one user has requested to receive;
 comparing retrieved information to previously retrieved information to identify one or more data changes and generating an event for each data change;
 accessing user criteria information that specifies one or more criteria for at least one user for which the user has requested to be notified upon the identification of a matching property data change;
 identifying one or more properties marked by at least one user for which changes to the properties are requested by the user to generate an alert; and
 scheduling alerts to notify users of generated events that match at least one accessed user criteria or marked property,
 wherein the preceding steps are performed by at least one processor.

2. The method of claim 1 wherein receiving the request comprises running a process on a tight frequency multiple times per hour that requests for new events to be imported.

3. The method of claim 1 wherein receiving the request comprises receiving information identifying one or more data sources for which new information is rapidly available.

4. The method of claim 1 wherein accessing event data sources comprises connecting to each source according to a protocol of the particular source, retrieving new data from that source, and storing the data in a local data store of aggregate real estate information.

5. The method of claim 1 wherein accessing event data sources comprises receiving information in a bulk format and separating the information into individual events.

6. The method of claim 1 wherein accessing event data sources comprises comparing two sets of data accessed at different times to identify changes, and identifying each change as an event.

7. The method of claim 1 wherein comparing data changes comprises storing results of retrieving information each time information is retrieved, and keeping a history of such information so that comparisons can be performed.

8. The method of claim 1 wherein generating an event comprises indicating a type of change and a value associated with the change that can later be used to generate a text-based description of the event.

9. The method of claim 1 wherein accessing user criteria information comprises accessing one or more stored user profiles that include saved property searches.

10. The method of claim 1 wherein accessing user criteria information comprises accessing one or more stored user profiles and comparing data therein to the identified one or more data changes to identify matches.

11. The method of claim 1 wherein identifying marked properties comprises identifying one or more properties identified as favorites, and providing alerts for any data changes in favorite properties.

12. The method of claim 1 wherein identifying marked properties comprises identifying one or more properties identified as anti-favorites, and suppressing alerts for any data changes in anti-favorite properties.

13. The method of claim 1 wherein scheduling alerts comprises invoking a process to send scheduled alerts to users after completing a process of identifying data changes.

14. A computer system for providing fast alerts for rapidly changing real estate data, the system comprising:

a processor and memory configured to execute software instructions embodied within the following components;
 a user profile component that stores user profile information for one or more users of the system;
 one or more event sources that include at least one multiple listing service (MLS);
 a real estate data store that stores information from the event sources and user profile information;
 a marked listing component that manages one or more listings marked by users as favorites;
 an event import component that detects events from the event sources and schedules the detected events for short-term notification to applicable users;
 a user alert component that periodically analyzes the real estate data store and detected events to identify events relevant to particular users and to communicate those events to the particular users; and
 an alert communication component that handles communication of identified alerts to each user.

15. The system of claim 14 wherein the user profile component stores one or more saved property searches that specify one or more criteria for properties of interest to users of the system.

16. The system of claim 14 wherein the marked listings component applies user favorite information to notifications to alert users to changes in properties that are marked as favorites.

17. The system of claim 14 wherein the marked listings component also tracks listings that a user has marked as non-favorites and suppresses notifications for changes in listings that are marked as non-favorites.

18. The system of claim 14 wherein the event import component detects data changes from the event sources multiple times per hour, and sends notifications to users in real time upon detecting a data change.

19. The system of claim 14 wherein the event import component adaptively schedules alerts for events based on event types, so that some events are scheduled for immediate notification to users and other events are delayed for notification on a longer-term schedule.

20. A computer-readable storage medium comprising instructions for controlling a computer system to provide alerts to one or more users of a real estate data system, wherein the instructions, upon execution, cause a processor to perform actions comprising:
 receiving a request to send pending alerts to users of the real estate data system;
 accessing one or more pending alert items previously detected based on one or more changes in data provided to the real estate data system by a real estate data provider;
 selecting a first user to which to provide alerts;
 identifying one or more pending alerts applicable to the selected user;
 identifying one or more communication channels requested by the user for receiving the pending alerts; and
 sending the identified alerts to the selected user via the identified communication channels, wherein the system alerts users rapidly without waiting for the expiration of a specified period or the occurrence of multiple events before sending information to the user.

* * * * *