



US 20150242421A1

(19) **United States**(12) **Patent Application Publication**
Glover et al.(10) **Pub. No.: US 2015/0242421 A1**(43) **Pub. Date: Aug. 27, 2015**(54) **LOCATION-BASED SEARCHING**(71) Applicant: **Quixey, Inc.**, Mountain View, CA (US)(72) Inventors: **Eric J. Glover**, Palo Alto, CA (US);
Michael Harris, Mountain View, CA (US); **James Delli Santi**, San Jose, CA (US)(73) Assignee: **Quixey, Inc.**, Mountain View, CA (US)(21) Appl. No.: **14/586,250**(22) Filed: **Dec. 30, 2014****Related U.S. Application Data**

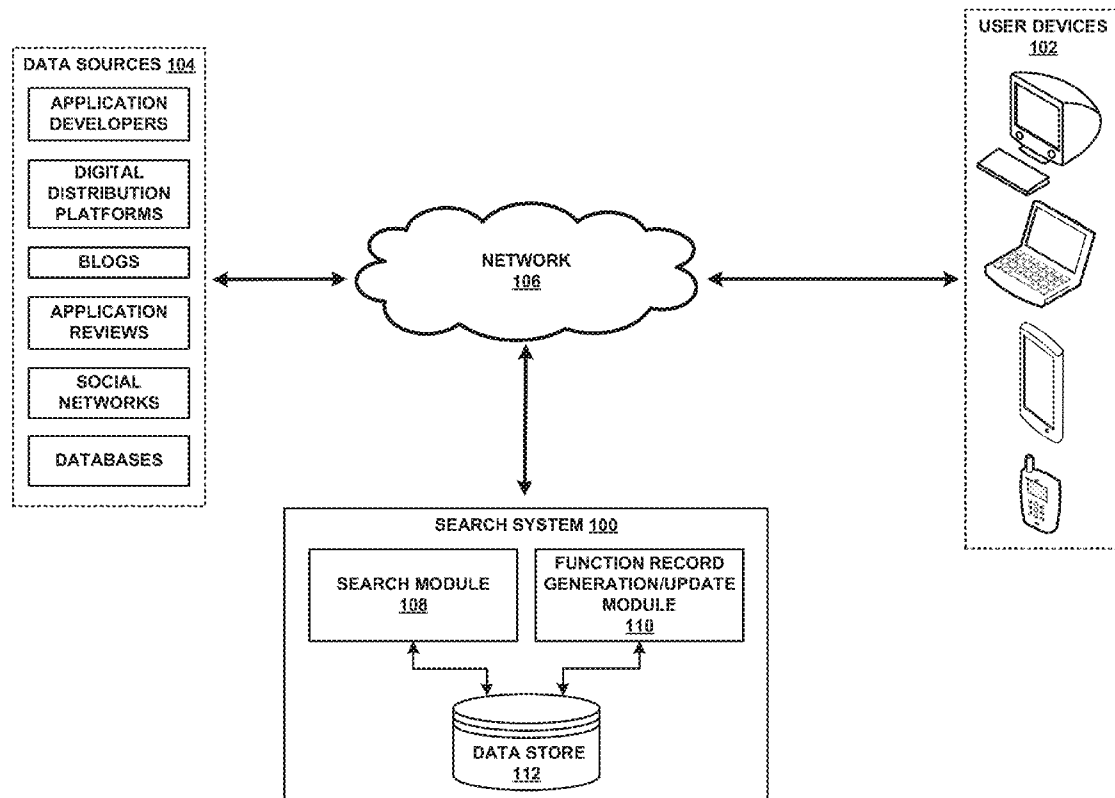
(60) Provisional application No. 61/943,105, filed on Feb. 21, 2014.

Publication Classification(51) **Int. Cl.**
G06F 17/30 (2006.01)(52) **U.S. Cl.**CPC **G06F 17/3087** (2013.01); **G06F 17/30241** (2013.01); **G06F 17/3053** (2013.01); **G06F 17/30867** (2013.01)

(57)

ABSTRACT

A method includes receiving a search query from a user device and identifying a plurality of function records included in a data store based on the received search query. Each of the function records includes an access mechanism specifying a state of an application, state information describing the state of the application, and location data including a coordinate and a perimeter. The coordinate defines the location of a place corresponding to the state information and the perimeter defines a geographic area surrounding the coordinate. The method also includes determining a search location and a score for the function record based on where the determined search location is located relative to the geographic area defined by the location data for each of the identified function records. The method also includes selecting access mechanisms from the function records based on the scores and transmitting the selected access mechanisms to the user device.



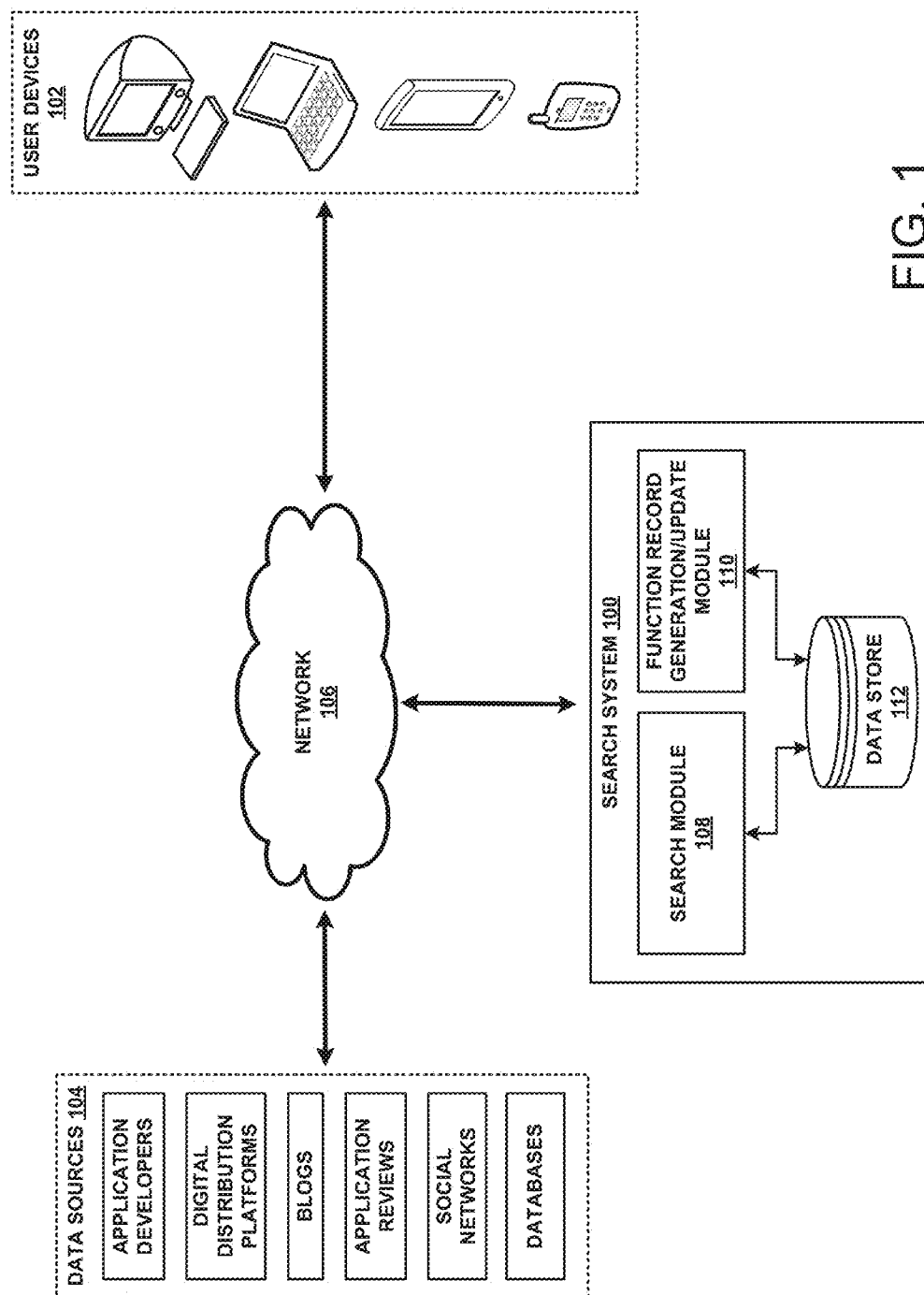


FIG. 1

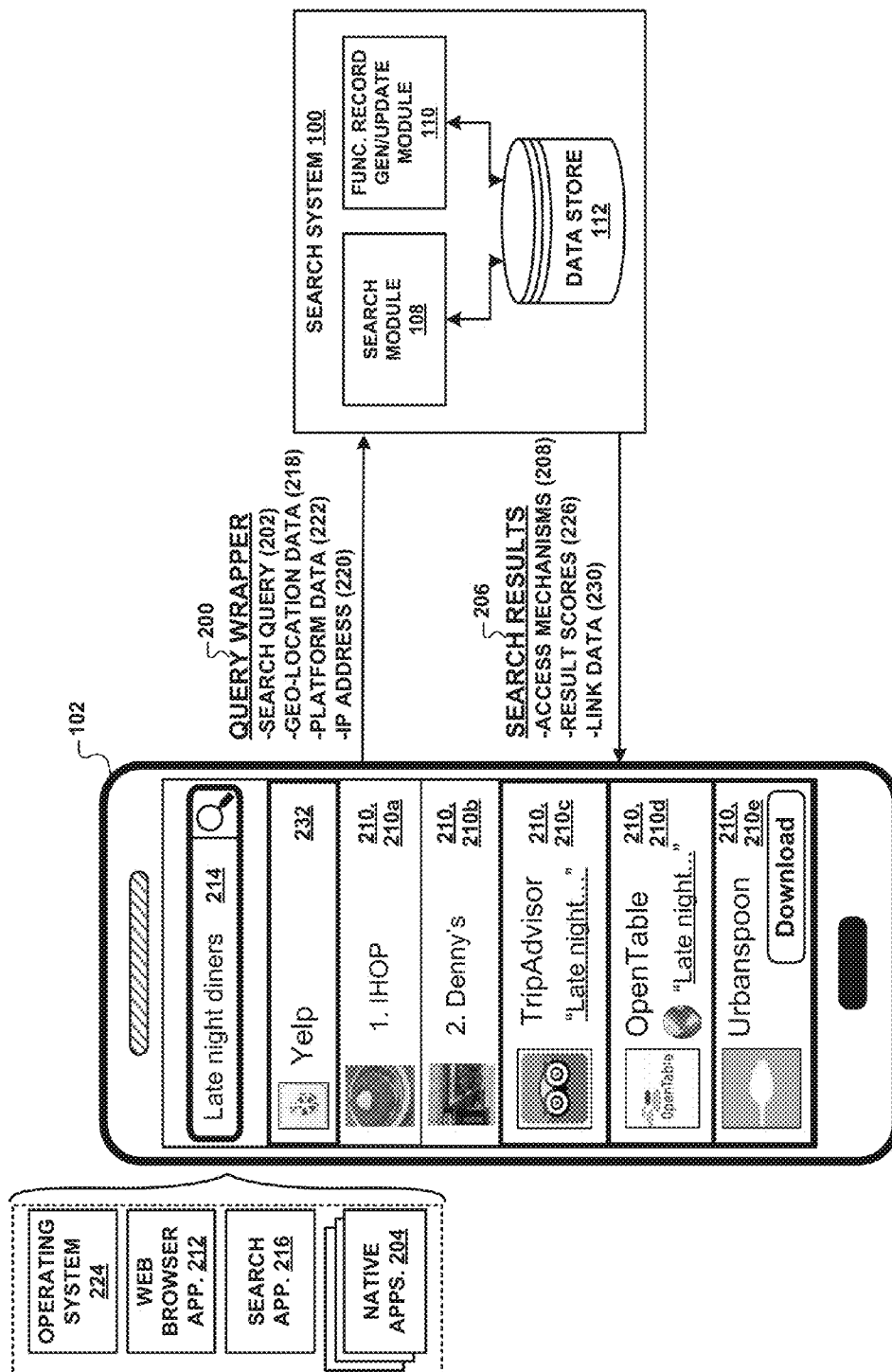


FIG. 2

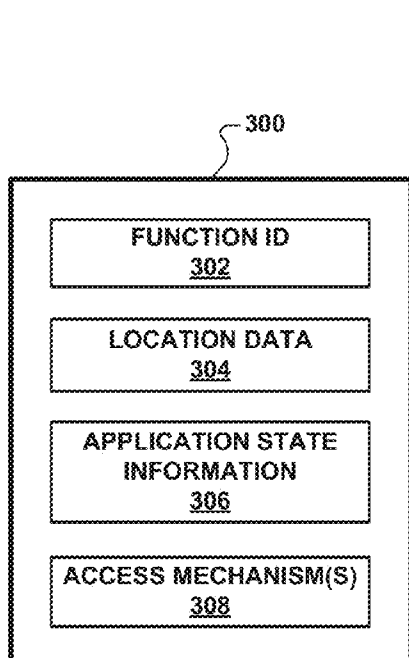


FIG. 3A

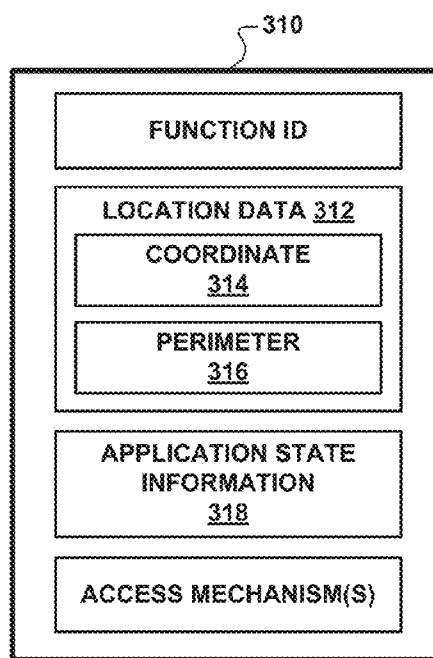


FIG. 3B

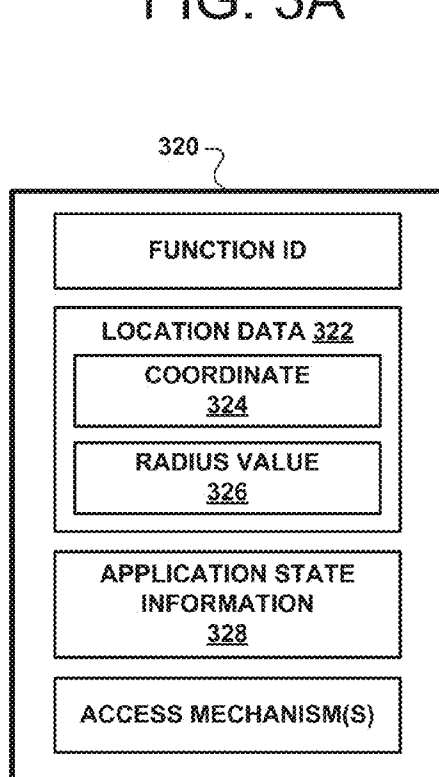


FIG. 3C

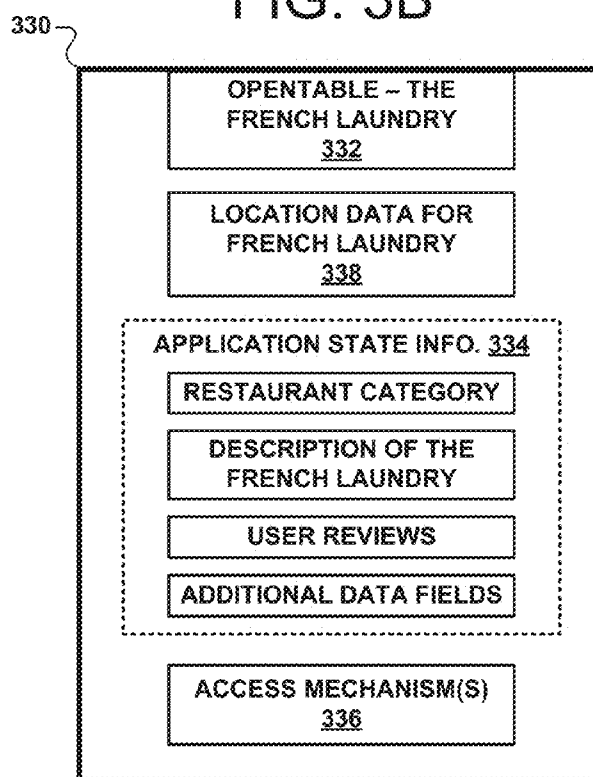


FIG. 3D

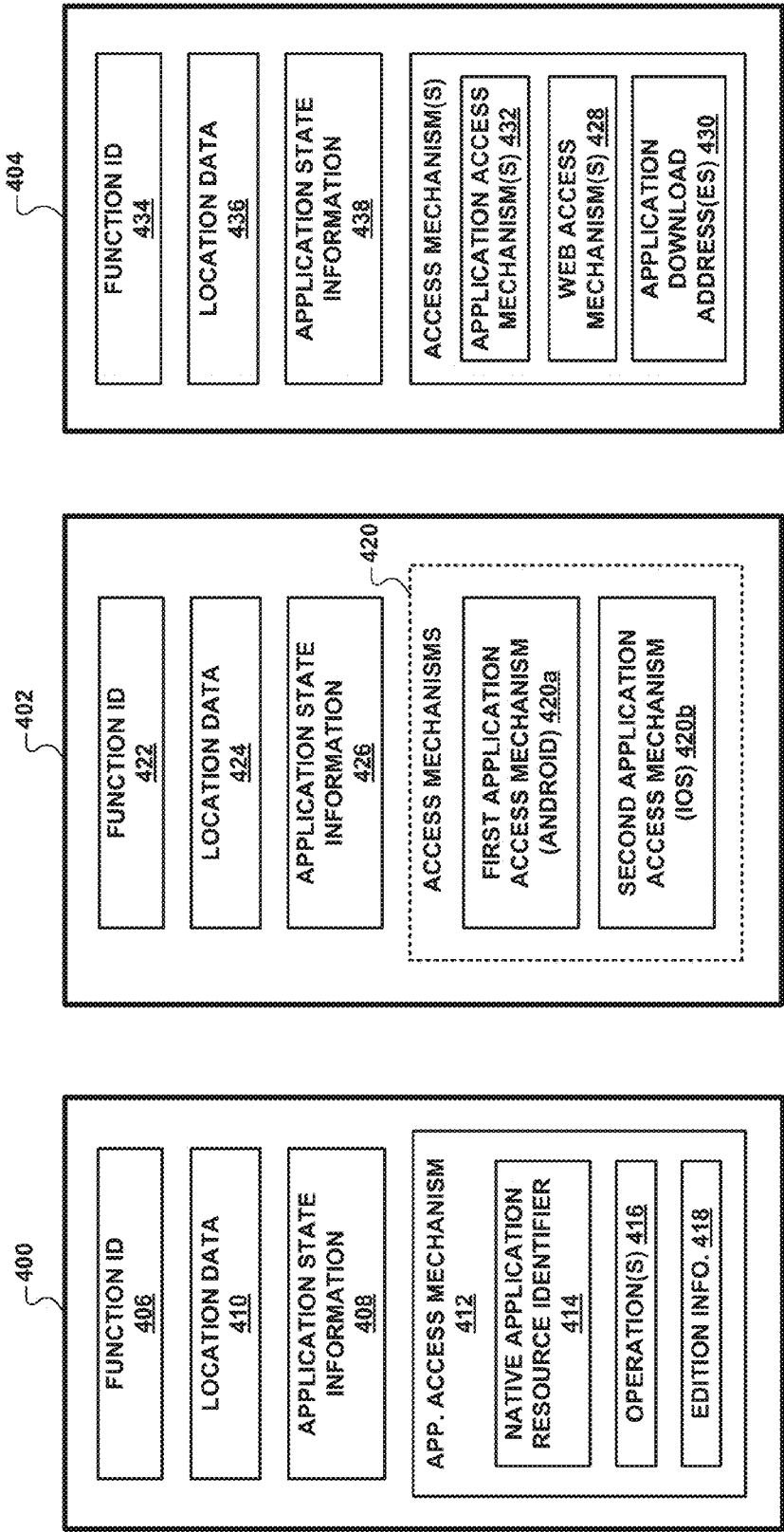


FIG. 4A

FIG. 4B

FIG. 4C

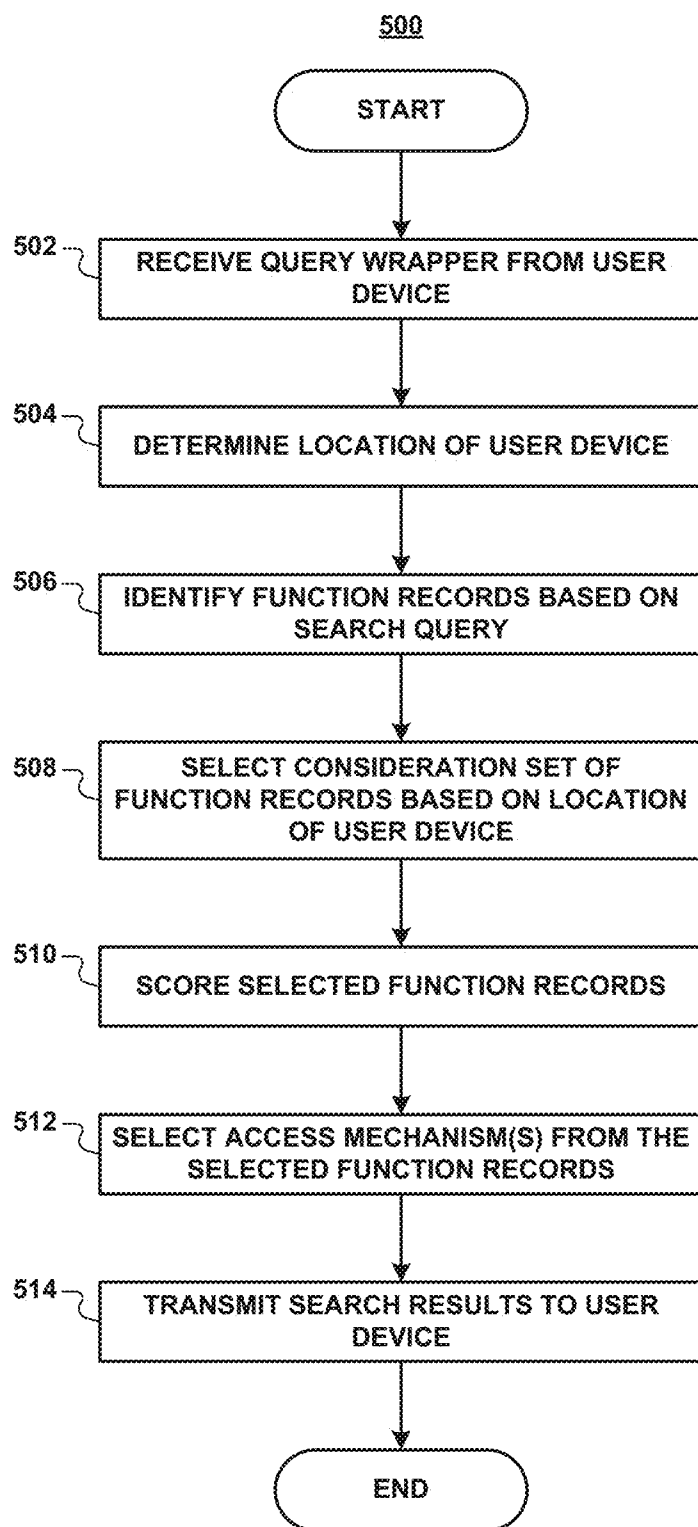


FIG. 5

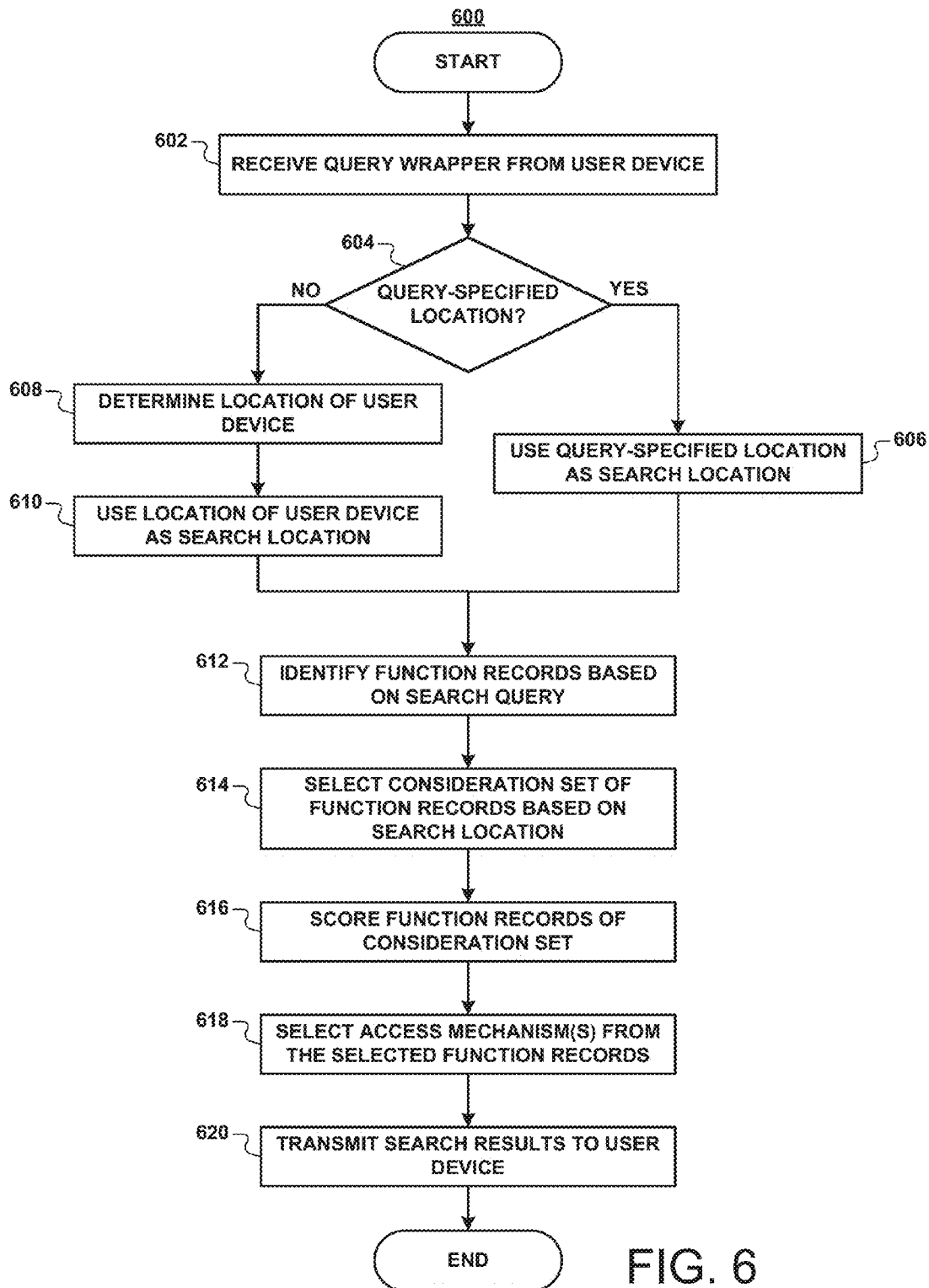


FIG. 6

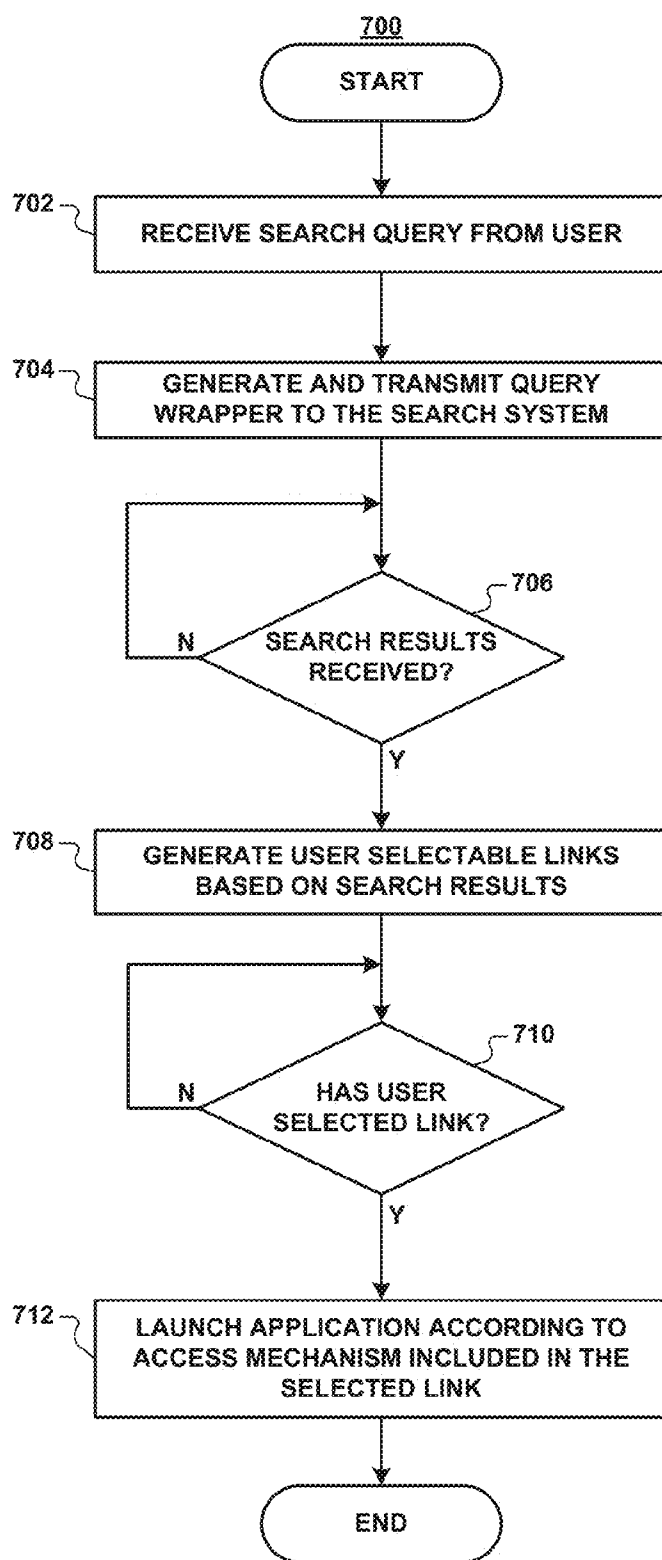


FIG. 7

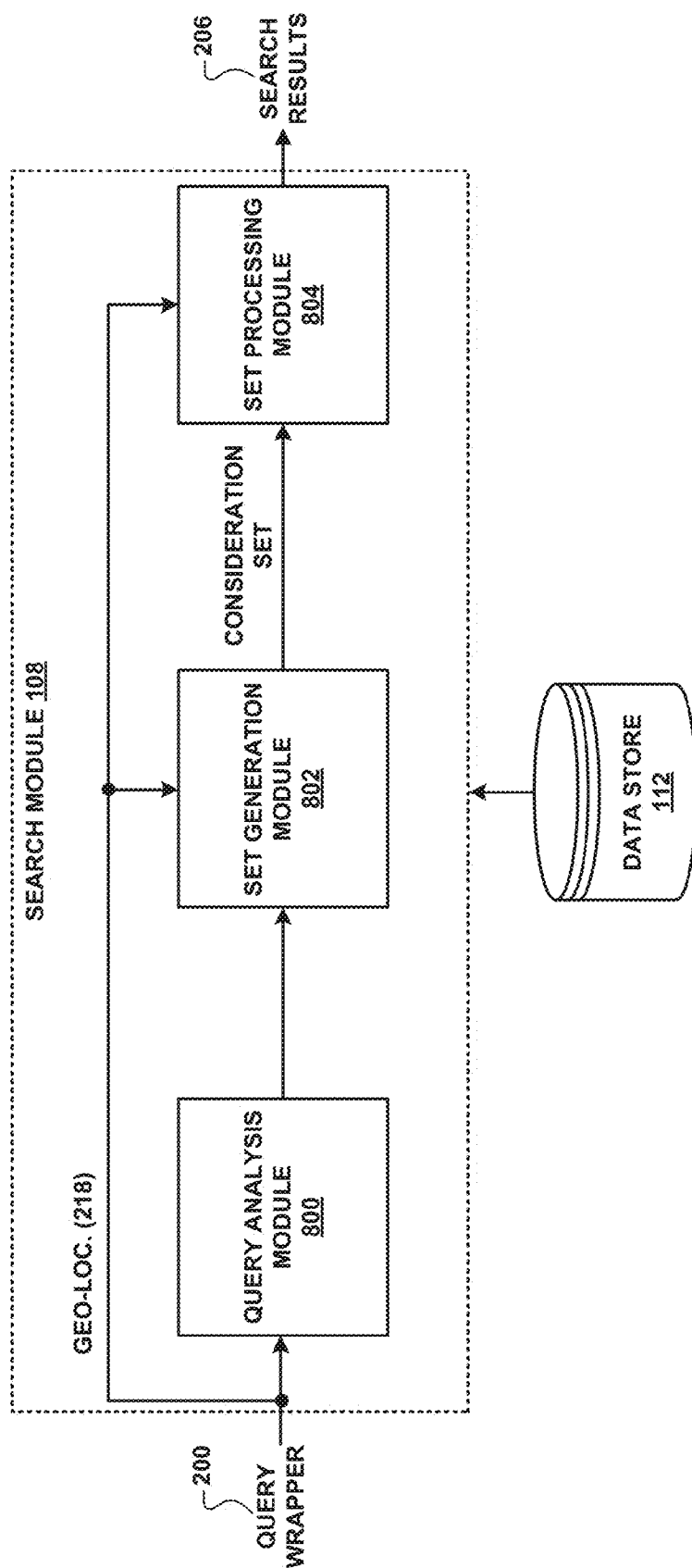


FIG. 8

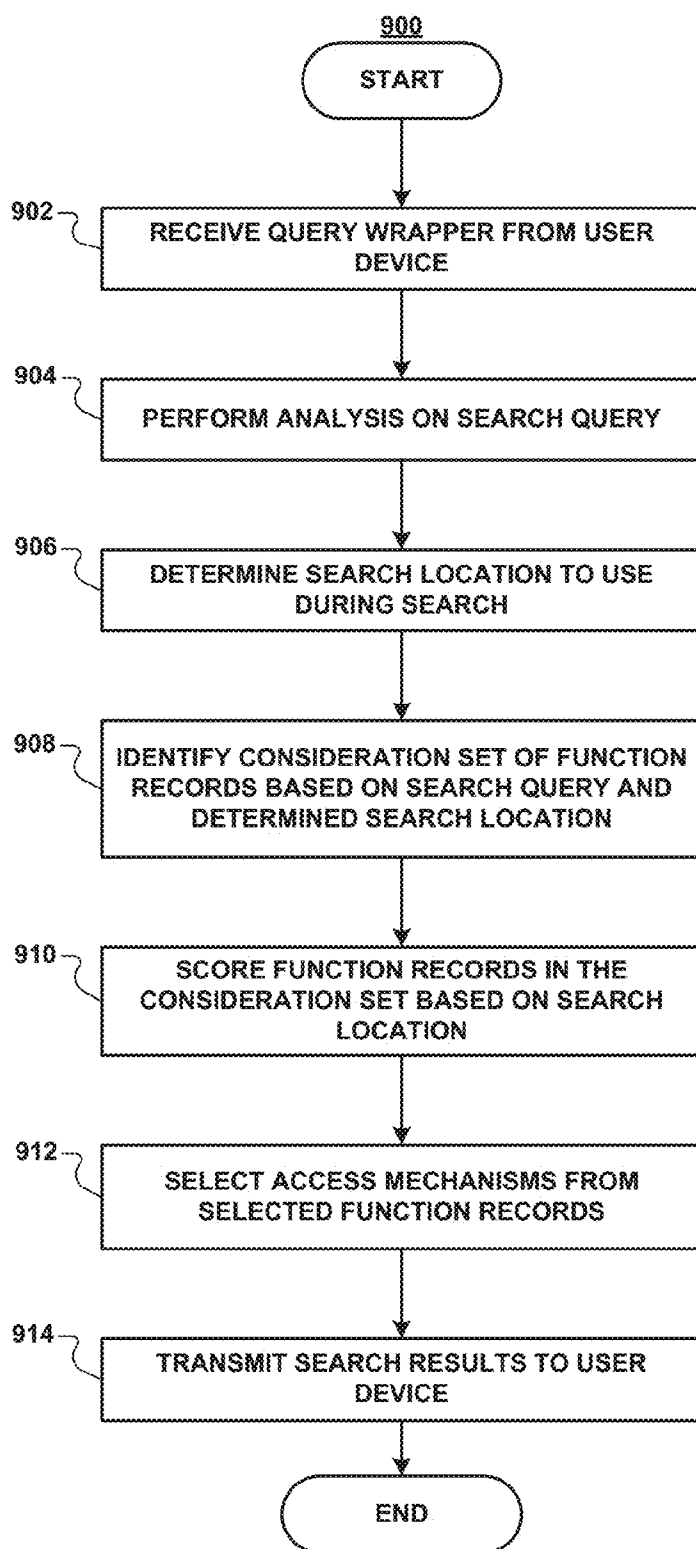


FIG. 9

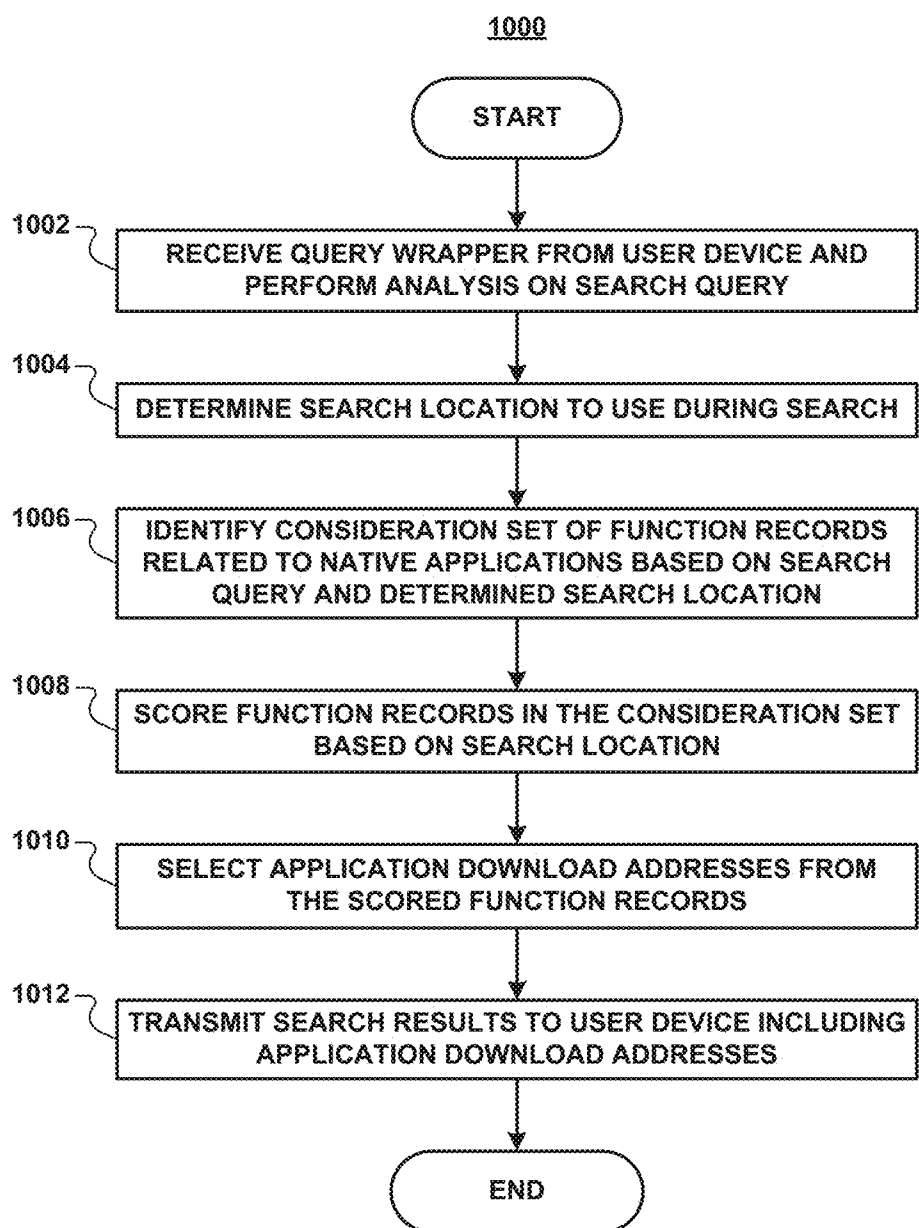


FIG. 10

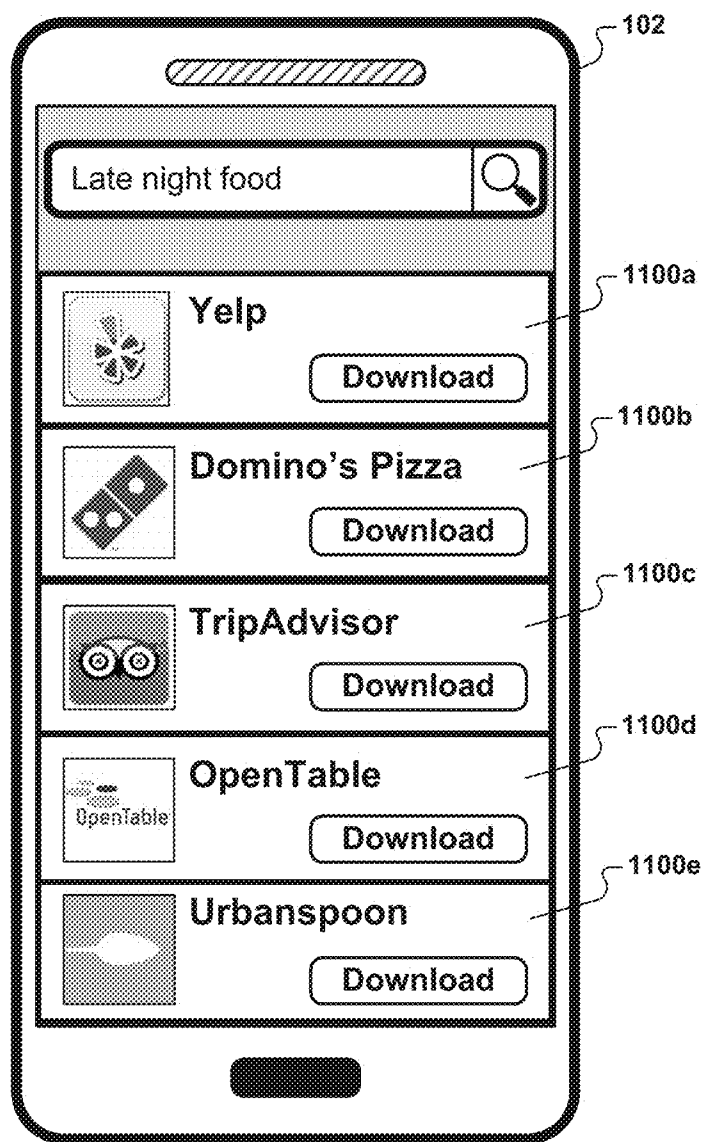


FIG. 11

LOCATION-BASED SEARCHING

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This U.S. application claims priority under 35 U.S.C. §119(e) to U.S. Provisional Application 61/943,105, filed on Feb. 21, 2014, which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

[0002] This disclosure relates to techniques for generating search results, and more particularly to generating search results based on a geographic location.

BACKGROUND

[0003] In recent years, use of computers, smartphones, and other Internet-connected devices has grown exponentially. Correspondingly, the number of available software applications for such devices has also grown. Today, many diverse native and web software applications can be accessed on any number of different devices, including, but not limited to, smartphones, personal computers, automobiles, and televisions. These diverse applications can range from business driven applications, games, educational applications, news applications, shopping applications, messaging applications, media streaming applications, social networking applications, and so much more. Furthermore, application developers develop vast amounts of applications within each genre and each application may have numerous editions.

SUMMARY

[0004] One aspect of the disclosure provides a method includes receiving a search query from a user device at a computing device and identifying, by the computing device, a plurality of function records included in a data store based on the received search query. Each of the function records includes an access mechanism specifying a state of an application, state information describing the state of the application specified by the access mechanism, and location data including a coordinate and a perimeter. The coordinate defines the location of a place corresponding to the state information and the perimeter defines a geographic area surrounding the coordinate. The method also includes determining, by the computing device, a search location, and determining, by the computing device, a score for the function record based on where the determined search location is located relative to the geographic area defined by the location data of the function record for each of the identified function records. The method also includes selecting, by the computing device, access mechanisms from the function records based on the scores associated with the function records, and transmitting the selected access mechanisms from the computing device to the user device.

[0005] Implementations of the disclosure may include one or more of the following optional features. In some implementations, the score indicates a relevance of the function record to the search query and the search location. Determining the score for the function record may include determining the score based on whether the search location is located within the geographic area defined by the perimeter. In some scenarios, determining the score for the function record includes determining a first score when the search location is located within the geographic area defined by the perimeter,

and determining a second score that is different than the first score when the search location is located outside of the geographic area defined by the perimeter. When the search location is located within the perimeter, the method may also include determining the score for the function record based on where the search location is located relative to the coordinate. Additionally or alternatively, when the search location is located outside of the perimeter, the method may also include determining the score for the function record based on the distance of the search location from the perimeter.

[0006] In some examples, the method further includes generating, by the computing device, the perimeter for at least one of the function records based on the population of the area surrounding the coordinate. Optionally, the method also includes generating, by the computing device, the perimeter for at least one of the function records based on the population density of the area surrounding the coordinate. In some scenarios, the method includes generating, by the computing device, the perimeter for at least one of the function records based on the proximity of the coordinate to transportation infrastructure. In some implementations, the method includes generating, by the computing device, the perimeter for at least one of the function records based on the popularity of the place defined by the state information.

[0007] In some implementations, for at least one of the plurality of function records, the perimeter is a circular perimeter defining a circular geographic area surrounding the coordinate and the coordinate defines a center point of the circular perimeter. In this implementation, the location data includes a radius value around the coordinate that defines the circular perimeter, and the method further includes generating the radius value based on the population of the geographic area surrounding the coordinate. In some scenarios, the perimeter is a polygon perimeter defining a geographic area surrounding the coordinate and the coordinate defines a center point of the polygon perimeter for at least one of the plurality of function records.

[0008] In some examples, for each of the plurality of function records, the method also includes determining, by the computing device, whether the search location is located within the geographic area defined by the location data of the function record, and filtering out, by the computing device, the function record if the determined search location is located outside of the geographic area defined by the location data of the function record. In this implementation, determining the score for the function record includes determining the score for the function record based on where the determined search location is located within the geographic area defined by the location data of the function record. In some scenarios, the method also includes receiving, at the computing device, data from the user device indicating the geo-location of the user device. In this scenario, determining the search location includes determining the search location based on at least one of a location specified in the search query and the data from the user device indicating the geo-location of the user device.

[0009] Another aspect of the disclosure provides a system including one or more computing devices and a data store that includes function records. Each function record includes an access mechanism specifying a state of an application, state information corresponding to the state of the application specified by the access mechanism, and location data including a coordinate and a perimeter. The coordinate defines the location of a place corresponding to the state information and the perimeter defines a geographic area surrounding the coor-

dinate. The one or more computing devices are configured to receive a search query from a user device, identify a plurality of function records included in the data store based on the received search query, and determine a search location. The one or more computing devices are further configured to determine a score for the function record based on where the determined search location is located relative to the geographic area defined by the location data of the function record, select access mechanisms from function records that include location based on the scores associated with the function records, and transmit the selected access mechanisms to the user device.

[0010] This aspect may include one or more of the following optional features. Optionally, the one or more computing devices are configured to generate the perimeter for at least one of the function records based on the population of the area surrounding the coordinate. In some examples, the one or more computing devices are configured to generate the perimeter for at least one of the function records based on the population density of the area surrounding the coordinate.

[0011] In sonic implementations, for at least one of the plurality of function records, the perimeter is a circular perimeter defining a circular geographic area surrounding the coordinate and the coordinate defines a center point of the circular perimeter. In this implementation, the location data includes a radius value around the coordinate that defines the circular perimeter, and the one or more computing devices are further configured to generate the radius value based on the population of the geographic area surrounding the coordinate.

[0012] In some examples, for each of the plurality of function records, the one or more computing devices are also configured to determine whether the search location is located within the geographic area defined by the location data of the function record for each of the plurality of function records, and filter out the function record if the determined search location is located outside of the geographic area defined by the location data of the function record. In this example, the one or more computing devices are configured to determine the score for the function record based on where the determined search location is located within the geographic area defined by the location data of the function record.

[0013] In yet another aspect of the disclosure provides a non-transitory computer-readable storage medium including instructions that cause one or more computing devices to receive a search query from a user device and identify a plurality of function records included in the data store based on the received search query, and determine a search location. Each function record includes an access mechanism specifying a state of an application, state information corresponding to the state of the application specified by the access mechanism, and location data including a coordinate and a perimeter. The coordinate defines the location of a place corresponding to the state information and the perimeter defines a geographic area surrounding the coordinate. The instructions also cause the one or more computing devices to determine a score for the function record based on where the determined search location is located relative to the geographic area defined by the location data of the function record, select access mechanisms from the function records based on the scores associated with the function records, and transmit the selected access mechanisms to the user device.

[0014] The details of one or more implementations of the disclosure are set forth in the accompanying drawings and the

description below. Other aspects, features, and advantages will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

[0015] FIG. 1 is a schematic view of an example environment including a search system.

[0016] FIG. 2 is a schematic view of an example user device in communication with a search system.

[0017] FIGS. 3A-4C are schematic views of example function records that include location data.

[0018] FIG. 5 is a flow diagram of an example method for performing a location-based search.

[0019] FIG. 6 is a flow diagram of another example method for performing a location-based search.

[0020] FIG. 7 is a flow diagram of an example method describing operation of a user device.

[0021] FIG. 8 is a functional block diagram of an example search module.

[0022] FIG. 9 is a flow diagram of another example method for performing a location-based search.

[0023] FIG. 10 is a flow diagram of an example method for performing a location-based search directed to application discovery.

[0024] FIG. 11 is a schematic view of an example graphical user interface (GUI) directed to application discovery.

[0025] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0026] A search system of the present disclosure implements a location-based search. The search system receives a query wrapper from a user device that includes a search query and data indicating the geo-location of the user device. The search system generates search results in response to the received query wrapper and transmits the search results to the user device. As described herein, the search system may generate the search results based on the location of the user device (i.e., the user's location) and/or a location specified in the search query so that search results are geographically relevant to the user. The location used by the search system (e.g., device or query-specified location) to generate the search results may be generally referred to herein as a "search location."

[0027] The search system stores records (referred to herein as "function records") that the search system may use to implement the location-based search of the present disclosure. Each function record may include one or more access mechanisms, application state information, and location data that the search system may use to generate the search results. The search system may transmit access mechanisms in the search results that the user device may use to access application functionality. Access mechanisms may include native application access mechanisms and web access mechanisms used to access functionality of native applications (e.g., installed on the user device) and web applications/websites, respectively. Access mechanisms may also include application download addresses that indicate sites (e.g., web/native) where a native application can be downloaded in the scenario where the native application is not installed on the user device.

[0028] The application state information of a function record includes data that describes an application state into

which an application is set according to the access mechanism(s) of the function record. For example, the application state information of a function record may include data that may be presented to the user by an application when the application is in the application state specified by the access mechanism(s) of the function record. In some examples, the application state information may include data that describes the function performed according to the access mechanism included in the function record.

[0029] The location data of a function record may define a geographic area that includes a place described in the application state information of the function record. The location data may include a coordinate (e.g., latitude and longitude) that indicates the location of the place. The location data may also define a perimeter that surrounds the place (e.g., coordinate). Put another way, the perimeter may define the geographic area that includes the place described in the application state information. The location data may define regular or irregular shapes. In one example, the location data may include a center point and a radius value that define a circular geographic area (e.g., having a circular perimeter) around the place described in the application state information.

[0030] In operation, the search system receives a query wrapper from the user device. The search system may determine a geo-location (e.g., latitude and longitude) of the user device based on data included in the query wrapper. The search system identifies a plurality of function records based on the received search query. For example, the search system may identify function records based on matches between terms of the search query and terms included in the application state information of the function records. The search system may filter and/or score the identified function records based on a determined search location and location data included in the function records.

[0031] In some implementations, the search system may select one or more function records in which the geo-location of the user device is included within the geographic area defined by the location data of the function records. The search system may then generate search results based on the selected function records. For example, the search system may include access mechanisms from the selected function records in the search results. The search system may refrain from transmitting access mechanisms (e.g., filter out access mechanisms) of function records in which the geo-location data of the user device is outside the geographic area defined by the location data.

[0032] The search system may generate scores for function records that indicate the relevance of the function record to the search query. In some implementations, the search system may generate scores for identified function records based on the geo-location of the user device and the location data of the identified function records. For example, the search system may generate a score for a function record based on where the user device is located relative to the geographic area defined by the location data. In one example, the search system may generate a score for a function record based on where the user device is located inside of the geographic area.

[0033] In some implementations, the search system may use a location other than that of the user device in order to filter and/or score the identified function records. For example, the search system may detect a location specified in the search query (i.e., a query-specified location) that the search system may use to filter and/or score the identified function records. The search system may use the geo-location

of the user device to filter and/or score the function records in the absence of a query-specified location. In other implementations, the search system may use a user-preferred location that is specified by the user (e.g., specified in the search application). In these implementations, the search system may use the user-preferred location to filter and/or score the function records instead of the geo-location of the user device.

[0034] The location data of a function record may be generated and updated in a variety of different ways. In some examples, a human operator may generate and/or update the location data. In other examples, the search system may automatically generate and/or update the location data of the function records. The search system (or a human operator) may generate the location data based on factors including, but not limited to, a population of a geographic area, a population density of a geographic area, a number of similar places that are nearby (e.g., similar types of entities, such as restaurant entities or dentist entities), a popularity of the place, a proximity of the place to transportation infrastructure (e.g., a highway), user behavior data indicating the distance users are willing to travel for a place, machine learned distance-related features (e.g., from keywords in the description), advertiser data (e.g., an advertiser specifying to advertise their business with a geo-fence, and user-interaction with ads. Generating and updating location data on a per-record basis allows the search system to tailor the geographic relevance of places independently from one another based on a variety of different factors, which may increase the probability that the search results are geographically relevant to the user.

[0035] In some examples, location data may include a coordinate (e.g., center point) and define a perimeter (e.g., a radius) that surrounds the coordinate. In these examples, the search system may generate and/or update the perimeter based on the above factors, or other factors. For example, the search system may generate a larger perimeter (e.g., larger radius) that covers a larger geographic area if the place is popular. As another example, the search system may generate a smaller perimeter (e.g., a smaller radius) if there are similar places nearby (e.g., other stores in the same chain).

[0036] Location-based searching techniques are described with reference to FIGS. 1-11. FIGS. 1-2 illustrate an environment that includes a user device that communicates with a search system. FIGS. 3A-4C illustrate example function records that include location data. FIGS. 5-7 illustrate example methods for performing location-based search. FIGS. 8-9 describe an example search module that implements location-based search. FIGS. 10-11 describe an example search system and user device directed to application discovery.

[0037] FIG. 1 is a functional block diagram illustrating an example environment including a search system 100 that communicates with user devices 102 and data sources 104 via a network 106. The network 106 through which the search system 100 and the user devices 102 communicate may include various types of networks, such as a local area network (LAN), wide area network (WAN), and/or the Internet. FIG. 2 shows an example user device 102 in communication with the search system 100 via the network 106 (not illustrated in FIG. 2).

[0038] The search system 100 includes a search module 108, a function record generation/update module 110 (hereinafter “record generation module 110”), and a data store 112. The search module 108 receives a query wrapper 200 from

the user device 102. The search module 108 performs a search for function records included in the data store 112 based on data included in the query wrapper 200, such as a search query 202. The function records include one or more access mechanisms that the user device 102 can use to access different functions for a variety of different applications, such as native applications 204 installed on the user device 102. The search module 108 transmits search results 206 including a list of access mechanisms 208 to the user device 102 that generated the query wrapper 200. As described herein, the record generation module 110 may generate new function records in the data store 112 and update existing function records.

[0039] The user device 102 generates user selectable links based on the received search results 206 (e.g., links 2 10a, 210b, . . . , 210e of FIG. 2). Each user selectable link displayed to the user may include an access mechanism. A user may select a user selectable link on the user device 102 by interacting with the link (e.g., touching or clicking the link). In response to selection of a link, the user device 102 may launch the application (e.g., native application or web application) referenced by the access mechanism and perform one or more operations indicated in the access mechanism.

[0040] A software application may refer to computer software that causes a computing device to perform a task. In some examples, a software application may be referred to as an “application”, an “app”, or a “program.” Example applications include, but are not limited to, word processing applications, spreadsheet applications, messaging applications, media streaming applications, social networking applications, and games.

[0041] Applications can be executed on a variety of different computing devices. For example, applications can be executed on mobile computing devices such as smart phones, tablets, and wearable computing devices (e.g., headsets and/or watches). Applications can also be executed on other types of computing devices having other form factors such as laptop computers, desktop computers, or other consumer electronic devices. In some examples, applications may be installed on a computing device prior to a user purchasing the computing device. In other examples, the user may download and install applications on the computing device.

[0042] The functionality of an application may be accessed on the computing device on which the application is installed. Additionally, or alternatively, the functionality of an application may be accessed via a remote computing device. In some examples, all of an application’s functionality is included on the computing device on which the application is installed. These applications may function without communication with other computing devices (e.g., via the Internet). In other examples, an application installed on a computing device may access information from other remote computing devices during operation. For example, a weather application installed on a computing device may access the latest weather information via the Internet and display the accessed weather information to the user through the installed weather application. In still other examples, an application (e.g., a web based application) may be partially executed by the user’s computing device and partially executed by a remote computing device. For example, a web application may be an application that is executed, at least in part, by a web server and accessed by a web browser of the user’s computing device. Example web applications may include, but are not limited to, web-based email, online auctions, and online retail sites.

[0043] Access mechanisms may include at least one of a native application access mechanism (hereinafter “application access mechanism”), a web access mechanism, and an application download address. The user device 102 may use the access mechanisms to access functionality of applications. For example, the user may select a user selectable link including an access mechanism in order to access functionality of an application indicated in the user selectable link. As described herein, the search module 108 may transmit one or more application access mechanisms, one or more web access mechanisms, and one or more application download addresses to the user device 102 in the search results 206.

[0044] An application access mechanism may be a string that includes a reference to a native application (e.g., one of native applications 204) and indicates one or more operations for the user device 102 to perform. If a user selects a user selectable link including an application access mechanism, the user device 102 may launch the native application referenced in the application access mechanism and perform the one or more operations indicated in the application access mechanism.

[0045] A web access mechanism may include a resource identifier that includes a reference to a web resource (e.g., a page of a web application/website). For example, a web access mechanism may include a uniform resource locator (URL) (i.e., a web address) used with hypertext transfer protocol (HTTP). If a user selects a user selectable including a web access mechanism, the user device 102 may launch the web browser application 212 and retrieve the web resource indicated in the resource identifier. Put another way, if a user selects a user selectable link including a web access mechanism, the user device 102 may launch the web browser application 212 and access a state (e.g., a page) of a web application/website. In some examples, web access mechanisms may include URLs for mobile-optimized sites and/or full sites.

[0046] An application download address may indicate a site (e.g., a digital distribution platform) where a native application can be downloaded in the scenario where the native application is not installed on the user device 102. If a user selects a user selectable link including an application download address, the user device 102 may access a digital distribution platform from which the referenced native application may be downloaded. The user device 102 may access a digital distribution platform using at least one of the web browser application 212 and one of the native applications 204.

[0047] The search module 108 is configured to receive a query wrapper 200 from the user device 102 via the network 106. A query wrapper 200 may include a search query 202. A search query 202 may include text, numbers, and/or symbols (e.g., punctuation) entered into the user device 102 by the user. For example, the user may have entered the search query 202 into a search field 214 (e.g., a search box) of a search application 216 running on the user device 102. A user may enter a search query using a touchscreen keypad, a mechanical keypad, and/or via speech recognition. As described herein, in some examples, the search application 216 may be a native application installed on the user device 102. For example, the search application 216 may receive search queries, generate the query wrapper 200, and display received data that is included in the search results 206. In other examples, the user device 102 may execute a web browser application 212 that accesses a web-based search application. In this example, the user may interact with the web-based

search application via the web browser application 212 installed on the user device 102. In still other examples, the functionality attributed to the search application 216 herein may be included as a searching component of a larger application that has additional functionality. For example, the functionality attributed to the search application 216 may be included as part of a native/web application as a feature that provides search for the native/web application.

[0048] The query wrapper 200 may include additional data along with the search query 202. For example, the query wrapper 200 may include geo-location data 218 that indicates the location of the user device 102, such as latitude and longitude coordinates. The user device 102 may include a global positioning system (GPS) receiver that generates the geo-location data 218 transmitted in the query wrapper 200. The query wrapper 200 may also include an IP address 220 that the search module 108 may use to determine the location of the user device 102. The query wrapper may also include additional data that the search module 108 may use to determine the location of the user device 102, such as data indicating current cell towers and nearby WiFi access points in communication with the user device 102. In some examples, the query wrapper 200 may also include additional data, including, but not limited to, platform data 222 (e.g., version of the operating system 224, device type, and web-browser version), an identity of a user of the user device 102 (e.g., a username), partner specific data, ISP/hostname, and other data, some of which may be used to determine the location of the user device 102.

[0049] The search module 108 can use the search query 202 and the additional data included in the query wrapper 200 to generate the search results 206. For example, the search module 108 can determine a geo-location of the user device 102 which the search module 108 can use along with the search query 202 to generate the search results 206. The search module 108 can determine the geo-location of the user device 102 based on the geo-location data 218 or other data (e.g., IP address 220) included in the query wrapper 200 to perform a location-based search. In some implementations, the search module 108 may detect a location (e.g., a postal address, street name, city name, etc.) that is specified in the search query 202 (i.e., a query-specified location). In these implementations, the search module 108 can use the query-specified location along with the search query 202 to generate the search results 206.

[0050] The search module 108 performs a search for function records included in the data store 112 in response to the received query wrapper 200 (e.g., search query 202 and geo-location data 218). In some implementations, the search module 108 generates result scores for function records identified during the search. The result score associated with a function record may indicate the relevance of the function record to the search query 202. A higher result score may indicate that the function record is more relevant to the search query 202. As described herein, the search module 108 may retrieve access mechanisms from the scored function records. The search module 108 can transmit a result score 226 along with an access mechanism retrieved from a scored function record in order to indicate the rank of the access mechanism among other transmitted access mechanisms 208.

[0051] The search module 108 may transmit additional data to the user device 102 along with the access mechanisms 208 and the result scores 226. For example, the search module 108 may transmit data (e.g., text and/or images) to be included in

the user selectable links. Data for the user selectable links (e.g., text and/or images) may be referred to herein as “link data” (e.g., link data 230). The user device 102 displays the user selectable links to the user based on received link data 230. Each user selectable link may be associated with an access mechanism included in the search results 206 such that when a user selects a link, the user device 102 launches the application referenced in the access mechanism and sets the application into the state specified by the access mechanism.

[0052] FIG. 2 shows an example list of user selectable links 210 that a user device 102 may display to a user. Each of the links 210 include link data. For example, each of the links 210 includes an image (e.g., an icon) and text (e.g., an application or business name) that may describe an application and a state of an application. Each of the links 210 may include an access mechanism so that if a user selects one of links 210, the user device 102 launches the application and sets the application into a state that is specified by the access mechanism associated with the selected link. In some implementations, the user device 102 may arrange the links 210 based on result scores associated with the access mechanisms included in the links 210. In some implementations, as illustrated in FIG. 2, links for the same application may be combined together in the search results displayed to the user.

[0053] User devices 102 can be any computing devices that are capable of providing search queries to the search system 100. User devices 102 include, but are not limited to, smart phones, tablet computers, laptop computers, and desktop computers. User devices 102 may also include other computing devices having other form factors, such as computing devices included in vehicles, gaming devices, televisions, or other appliances (e.g., networked home automation devices and home appliances).

[0054] The user devices 102 may use a variety of different operating systems. In an example where a user device 102 is a mobile device, the user device 102 may run an operating system including, but not limited to, ANDROID® developed by Google Inc., IOS® developed by Apple Inc., or WINDOWS PHONE® developed by Microsoft Corporation. Accordingly, the operating system 224 running on the user device 102 may include, but is not limited to, one of ANDROID®, IOS®, or WINDOWS PHONE®. In an example where a user device is a laptop or desktop computing device, the user device may run an operating system including, but not limited to, MICROSOFT WINDOWS® by Microsoft Corporation, MAC OS® by Apple, Inc., or Linux. User devices 102 may also access the search system 100 while running operating systems other than those operating systems described above, whether presently available or developed in the future.

[0055] In general, a user device 102 may communicate with the search system 100 using any application that can transmit search queries to the search system 100. In some examples, a user device 102 may run a native application that is dedicated to interfacing with the search system 100, such as a native application dedicated to searches (e.g., search application 216). In some examples, a user device 102 may communicate with the search system 100 using a more general application, such as a web-based application accessed using the web browser application 212. Although the user device 102 may communicate with the search system 100 using a web based application and/or a native search application, the user device 102 may be described hereinafter as using the native search application 216 to communicate with the search system 100.

[0056] The search application 216 may display a search field 214 on a graphical user interface (GUI) in which the user can enter search queries. The user may enter a search query using a touchscreen or physical keyboard, a speech-to-text program, or other form of user input. In general, a search query may be a request for information retrieval (e.g., search results) from the search system 100. For example, a search query may be directed to retrieving a list of links to application functionality or application states in examples where the search system 100 is configured to generate a list of access mechanisms as search results. A search query directed to retrieving a list of links to application functionality may indicate a user's desire to access functionality of one or more applications described by the search query.

[0057] A user device 102 may receive a set of search results 206 from the search module 108 that are responsive to the query wrapper 200 transmitted to the search system 100. The GUI of the search application 216 displays (e.g., renders) the search results 206 received from the search module 108. The search application 216 may display the search results 206 to the user in a variety of different ways, depending on what information is transmitted to the user device 102. In examples where the search results 206 include a list of access mechanisms and link data, the search application 216 may display the search results to the user as a list of user selectable links including text and images. The text and images in the links may include application names associated with the access mechanisms, text describing the access mechanisms, images associated with the application referenced by the access mechanisms (e.g., application icons), and images associated with the application state (e.g., application screen images) defined by the access mechanisms.

[0058] In some implementations, the search application 216 may display the search results 206 as a list of links (e.g., links 210 of FIG. 2) arranged under the search field (e.g., search field 214 of FIG. 2) in which the user entered the search query. The search application 216 may arrange the links in order by result scores associated with the access mechanisms included in the links. In some examples, the search application 216 may group the links together if the links are related to the same application.

[0059] With respect to FIG. 2, it may be assumed that the YELP® native application developed by Yelp, Inc., and the TRIPADVISOR® native application developed by TripAdvisor, Inc., are installed on the user device 102. Links 210a, 210b and link 210c reference the YELP® native application and the TRIPADVISOR® native application, respectively. The GUI includes a header 232, including the name “Yelp,” under which the links 210a, 210b are arranged. The header 232 may indicate that the links 210a, 210b arranged below the header 232 are associated with the YELP® native application. Selection of link 210a may cause the user device 102 to launch the YELP® native application and retrieve an IHOP® restaurant entry of the YELP® native application. Selection of link 210b may cause the user device 102 to launch the YELP® native application and retrieve a DENNY'S® restaurant entry of the YELP® native application. Selection of link 210c may cause the user device 102 to launch the TRIPADVISOR® native application and retrieve an entry for “Late night diners” in the TRIPADVISOR® native application (e.g., a search for “Late night diners”).

[0060] Link 210d includes a web access mechanism (e.g., a UPI). Selection of link 210d may cause the user device 102 to launch the web browser application 212 and retrieve an entry

for “Late night diners” in the OPENTABLE® web application developed by OpenTable, Inc. Link 210e includes an application download address for the URBANSPOON® native application by InterActiveCorp. Selection of link 210e may cause the user device 102 to access a digital distribution platform from which the URBANSPOON® native application can be downloaded and/or previewed. As described herein, the search module 108 can be configured to transmit any combination of application access mechanisms, web access mechanisms, and application download addresses in the search results 206.

[0061] In some examples, user devices 102 may communicate with the search system 100 via a partner computing system (not illustrated). The partner computing system may be a computing system of a third party that may leverage the search functionality of the search system 100. The partner computing system may belong to a company or organization other than that which operates the search system 100. Example third parties that may leverage the functionality of the search system 100 may include, but are not limited to, internet search providers and wireless communications service providers. The user devices 102 may send search queries to the search system 100 and receive search results via the partner computing system. The partner computing system may provide a user interface to the user devices 102 in some examples and/or modify the search experience provided on the user devices 102.

[0062] The data store 112 includes a plurality of different function records. Each function record may include data related to a function of an application and/or the state of the application resulting from performance of the function. A function record may include a function identifier (ID), location data, application state information, and one or more access mechanisms used to access functionality provided by an application. The data store 112 may include one or more databases, indices (e.g., inverted indices), tables, files, or other data structures which may be used to implement the techniques of the present disclosure. The search module 108 receives a query wrapper 200 and generates search results based on the data included in the data store 112.

[0063] FIG. 1 shows a plurality of data sources 104. The data sources 104 may be sources of data which the search system 100 (e.g., the record generation module 110) may use to generate and update the data store 112. The record generation module 110 retrieves data from one or more of the data sources 104. The data retrieved from the data sources 104 can include any type of data related to application functionality and/or application states. The record generation module 110 may use the data retrieved from the data sources 104 to create and/or update one or more databases, indices, tables (e.g., an access table), files, or other data structures included in the data store 112. For example, the record generation module 110 may create new function records and update existing function records based on data retrieved from the data sources 104. In some examples, some data included in the data store 112 may be manually generated by a human operator. For example, some data included in the function records (e.g., application state information) may be manually generated by a human operator. The record generation module 110 (or a human operator) may update the data included in the function records over time so that the search system 100 provides up-to-date results.

[0064] The data sources 104 may include a variety of different data providers. The data sources 104 may include data

from application developers, such as application developers' websites and data feeds provided by developers. The data sources **104** may include operators of digital distribution platforms configured to distribute native applications to user devices **102**. Example digital distribution platforms include, but are not limited to, the GOOGLE PLAY® digital distribution platform by Google, Inc., the APP STORE® digital distribution platform by Apple, Inc., and WINDOWS PHONE® Store developed by Microsoft Corporation.

[0065] The data sources **104** may also include other websites, such as websites that include web logs (i.e., blogs), application review websites, or other websites including data related to applications. Additionally, the data sources **104** may include social networking sites, such as "FACEBOOK®" by Facebook, Inc. (e.g., Facebook posts) and "TWITTER®" by Twitter Inc. (e.g., text from tweets). Data sources **104** may also include online databases that include, but are not limited to, data related to movies, television programs, music, and restaurants. Data sources **104** may also include additional types of data sources in addition to the data sources described above. Different data sources may have their own content and update rate.

[0066] Referring now to FIG. 3A, an example function record **300** includes a function identifier **302** (hereinafter "function ID **302**"), location data **304**, application state information **306**, and one or more access mechanisms **308**. The function record **300** may include data related to a function of an application and/or the state of the application resulting from performance of the function. The data store **112** may include a plurality of function records having a similar structure as the function record **300**. Put another way, the data store **112** may include a plurality of function records having a function ID **302**, location data **304**, application state information **306**, and one or more access mechanisms **308**.

[0067] The function ID **302** may be used to identify the function record **300** among other function records included in the data store **112**. The function ID **302** may be a string of alphabetic, numeric, and/or symbolic characters (e.g., punctuation marks) that uniquely identify the function record **300** in which the function ID **302** is included. In some examples, the function ID **302** may describe a function and/or an application state in human readable form. For example, the function ID **302** may include the name of the application referenced in the access mechanism(s) **308**. Additionally, or alternatively, the function ID **302** may be a human readable string that describes a function performed according to the access mechanism(s) **308** and/or an application state resulting from performance of the function according to the access mechanism(s) **308**. In some examples, the function ID **302** may include a string in the format of a uniform resource locator (URL) of a web access mechanism for the function record **300**, which may uniquely identify the function record.

[0068] In a more specific example, if the function record **300** describes a function of the YELP® native application, the function ID **302** may include the name "Yelp" along with a description of the application state described in the application state information **306**. For example, the function ID **302** for a function record that describes the restaurant named "The French Laundry" may be "Yelp—The French Laundry." In an example where the function ID **302** includes a string in the format of a URL, the function ID **302** may include the following string "http://www.yelp.com/biz/the-french-laundry-yountville-2?ob=1" to uniquely identify the function record **300**. In other examples, the function ID **302** may include a

URL using a namespace other than "http:," such as "func:/" which may indicate that the URL is being used as a function ID in a function record. For example, the function ID **302** may include the following string "func:www.yelp.com/biz/the-french-laundry-yountville-2?ob=1."

[0069] The function record **300** includes one or more access mechanisms **308**. The access mechanism(s) **308** may include one or more application access mechanisms, one or more web access mechanisms, and one or more application download addresses. The user device **102** may use the one or more application access mechanisms and the one or more web access mechanisms to access the same, or similar, functionality of the native/web application referenced in the application state information **306**. For example, the user device **102** may use the different access mechanism(s) **308** to retrieve similar information, play the same song, or play the same movie. The application download addresses may indicate sites (e.g., web/native, such as the GOOGLE PLAY® digital distribution platform) where the native applications referenced in the application access mechanisms can be downloaded.

[0070] The application state information **306** may include data that describes an application state into which an application is set according to the access mechanism(s) **308** in the function record **300**. Additionally, or alternatively, the application state information **306** may include data that describes the function performed according to the access mechanism(s) **308** included in the function record **300**. The application state information **306** may include a variety of different types of data. For example, the application state information **306** may include structured, semi-structured, and/or unstructured data. In some implementations, the record generation module **110** may extract and/or infer the application state information **306** from documents retrieved from the data sources **104**. Additionally, or alternatively, the application state information **306** may be manually generated data. The record generation module **110** may update the application state information **306** so that up-to-date search results can be provided in response to a search query.

[0071] In some examples, the application state information **306** may include data that may be presented to the user by an application when the application is set in the application state defined by the access mechanism(s) **308**. For example, if one of the access mechanism(s) **308** is an application access mechanism (e.g., **412** of FIG. 4A), the application state information **306** may include data that describes a state of the native application after the user device **102** has performed the one or more operations indicated in the application access mechanism. In one example, if the function record **300** is associated with a shopping application, the application state information **306** may include data that describes products (e.g., names and prices) that are shown when the shopping application is set to the application state defined by the access mechanism(s) **308**. As another example, if the function record **300** is associated with a music player application, the application state information **306** may include data that describes a song (e.g., name and artist) that is played when the music player application is set to the application state defined by the access mechanism(s) **308**.

[0072] The types of data included in the application state information **306** may depend on the type of information associated with the application state and the functionality defined by the access mechanism(s) **308**. In one example, if the function record **300** is for an application that provides reviews of

restaurants, the application state information **306** may include information (e.g., text and numbers) related to a restaurant, such as a category of the restaurant, reviews of the restaurant, and a menu for the restaurant. In this example, the access mechanism(s) **308** may cause the application (e.g., a web or native application) to launch and retrieve information for the restaurant (e.g., using the web browser application **212** or one of native applications **204**). As another example, if the function record **300** is for an application that plays music, the application state information **306** may include information related to a song, such as the name of the song, the artist, lyrics, and listener reviews. In this example, the access mechanism(s) **308** may cause the application to launch and play the song described in the application state information **306**.

[0073] The function record **300** includes location data **304**. In some implementations, the search module **108** may use the location data **304** to filter out function records that may not be relevant to a user because of the user's location relative to the place associated with the function records. For example, at search time, the search module **108** may filter out function records associated with places that are located too far from the user. Additionally, or alternatively, the search module **108** may score and rank the search results **206** based on the location data **304** included in the function records. For example, the search module **108** may assign larger result scores to function records associated with places that are nearer to the user.

[0074] The location data **304** may define a geographic area associated with the function record **300**. For example, the geographic area defined by the location data **304** may be a geographic area that includes a place described in the application state information **306**. In some examples, the geographic area defined by the location data **304** may define a perimeter that surrounds a geographic area including a place described in the application state information **306**. For example, if the access mechanism(s) **308** provide access to reviews of a specific restaurant in Mountain View, Calif., then the location data **304** may define a geographic area that surrounds the specific restaurant. As described above, the application state information **306** includes data that describes an application state into which an application is set according to the one or more access mechanism **308**. Accordingly, the location data **304** may also define a geographic area that surrounds a place for which the access mechanism(s) **308** retrieve information.

[0075] The geographic area defined by the location data **304** may have a variety of different geometries. For example, the location data **304** may define geographic area having a circular geometry, a polygon geometry (e.g., a rectangular geometry, hexagonal geometry, octagonal geometry, etc.), or other basic shape. In some examples, the location data **304** may define a geographic area having a more irregular shape. The location data **304** may define the geographic area in a variety of different ways, depending on the shape of the geographic area. In some examples, the location data **304** may include geographic coordinates (e.g., latitude and longitude) that define the geographic area. In other examples, the location data **304** may include one or more addresses (e.g., a postal address) that define the geographic area.

[0076] FIG. 3B shows another example function record **310**. Function record **310** includes example location data **312** that includes a coordinate **314** and a perimeter **316**. The coordinate **314** and the perimeter **316** together define a geo-

graphic area that surrounds the place described in the application state information **318**. The coordinate **314** may be a geographic coordinate of the place described in the application state information **318**. For example, the coordinate **314** may include a latitude value and a longitude value of the place described in the application state information **318**. In some examples, the coordinate **314** may include different data that indicates a location, such as an address (e.g., a postal address) of the place. In some implementations, the location data **304**, **312**, **322** may be generated using geocoding processes.

[0077] The perimeter **316** may surround the coordinate **314**. Accordingly, the area surrounding the coordinate **314** may be bounded by the perimeter **316**. The perimeter **316** can have a variety of different geometries. In some examples, the perimeter **316** may have a circular geometry, a polygon geometry (e.g., a rectangular geometry), or other basic shape. In other examples, the perimeter **316** may have a more irregular shape.

[0078] FIG. 3C illustrates an example function record **320** including location data **322** that defines a geographic area. The location data **322** includes a coordinate **324** and a radius value **326**. The coordinate **324** and the radius value **326** together define a circular geographic area that surrounds the place described in the application state information **328**. The coordinate **324** may be a geographic coordinate of the place described in the application state information **328**. The radius value **326** may define a circular perimeter that surrounds the coordinate **324**. For example, the perimeter around the coordinate **324** may be a circle having a center point at the coordinate **324** and a radius having the radius value **326**. Accordingly, the radius value **326** may define the radius length of a circular geographic area including the place described in the application state information **328**.

[0079] Although FIG. 3C illustrates an example where location data **322** defines a circular geographic area, function records may include location data that define geographic areas having other shapes. For example, a function record may include location data that defines a polygon region, such as a rectangular (e.g., square) region that surrounds the place described in the application state information. In this example, the coordinate may define the location of the place. The coordinate may be located at the center of the polygon region and the location data may define the dimensions of the polygon region (e.g., the side lengths). In examples where the polygon region is a rectangular region, the perimeter may be defined by a length value and a width value that define the length and width of the rectangular region surrounding the coordinate. In some examples, the coordinate may be located in the center of the rectangular region. In examples where the rectangular region has a square geometry, the perimeter may be defined by a single value that indicates the length of each side of the square region.

[0080] The size of the geographic area defined by the location data may vary. In some examples, the geographic area may be less than a square mile in area for those places that are more relevant to a user when the user is in close proximity to the place (e.g., less than 1 mile). With respect to location data **322**, the radius value **326** may be a fraction of a mile in length (e.g., approximately 0.5 miles) in these examples. In other examples, the geographic area may cover hundreds of square miles for places that are more relevant at greater distances from the user. With respect to location data **322**, the radius value **326** may be multiple miles in length (e.g., approximately 10 miles) in these examples. In still other examples,

the geographic area may have worldwide coverage for those places that are relevant to a user from any distance. With respect to location data 322, the radius value 326 may be thousands of miles in length, such that the geographic area covers most or all of the planet.

[0081] The location data 304 may be generated in a variety of different ways. In some implementations, the location data may be manually generated by a human operator. For example, a human operator may manually generate the initial location data (e.g., the coordinate 324 and the radius value 326). After generating the initial location data, the human operator may update the location data 304 at a later time. For example, with respect to function record 320, the human operator may update the coordinate 324 if the place described by the function record 320 moves to a different location. As another example, the human operator may update the perimeter 316 (e.g., the radius value 326) to reflect other changes, such as a population of a geographic area, a population density, a number of similar places that are nearby, and a popularity of the place. A user may manually update the location data 304 over time so that the search module 108 provides up-to-date results.

[0082] Although a human operator may generate and/or update the location data 304, in some implementations, the record generation module 110 may generate and/or update the location data 304. For example, the record generation module 110 may automatically generate the initial location data (e.g., the coordinate 324 and radius value 326) and update the location data 304 at a later time. The record generation module 110 may generate and/or update the location data 304 based on a variety of different factors. For example, the record generation module 110 may generate and update location data 304 based on data retrieved from data sources 104, or from other sources. In one example, the record generation module 110 may update the coordinate 314 if the place described by the function record 310 moves to a different location. As another example, the record generation module 110 may update the perimeter 316 to reflect other changes, such as a population of a geographic area, a population density, a number of similar places that are nearby, and a popularity of the place. The record generation module 110 may automatically update the location data 304 over time so that the search module 108 provides up-to-date results. Different techniques for generating the location data 304 are described herein.

[0083] The record generation module 110 (or a human operator) may generate the location data 304 based on a variety of different parameters. In some implementations, the record generation module 110 may generate the location data 304 based on the popularity of the place described by the application state information 306. The record generation module 110 may determine the popularity of a place based on ratings, reviews, data indicating how often the function record 300 is retrieved during prior searches, and/or how often access mechanism(s) 308 of the function record are selected by a user (e.g., the number of times access mechanism(s) 308 were selected when included in prior search results). In some examples, the record generation module 110 may generate location data 304 that defines a larger geographic area for places that are more popular. For example, the record generation module 110 may generate a larger radius value 326 in the location data 322 when the place described by the application state information 328 is popular. Increasing the size of the geographic area defined in the

location data 304 based on popularity may reflect the notion that more popular places may be more relevant to a user, even if the user is located farther from the place.

[0084] In some implementations, the record generation module 110 may generate the location data 304 based on the population and/or population density of the area in which the place described by the application state information 306 is located. In some examples, the record generation module 110 may generate location data 304 that defines a larger geographic area for places that are located in less populated places or places having less population density. For example, the record generation module 110 may generate a larger radius value 326 in the location data 322 when the place described by the application state information 328 is located in a low population area or an area that has a low population density. Increasing the size of the geographic area defined in the location data 304 based on population and/or population density may reflect the notion that places located farther away from a user may still be relevant to the user since users living in more sparsely populated areas may be required to travel greater distances to acquire more sparsely available goods and services.

[0085] In a specific example related to the generation of location data based on population density, it may be assumed that function records exist for four different WENDY'S® restaurants in the data store 112. Three of the WENDY'S® restaurants are located in a densely populated area in San Francisco, Calif., USA. The other WENDY'S® restaurant is located in a less densely populated area in Rock Springs, Wyo., USA. The function records associated with the three WENDY'S® restaurants in San Francisco may have a smaller geographic area (e.g., perimeter) associated with them than the function record for the WENDY'S® restaurant in Rock Springs, Wyo. For example, the record generation module 110 (or a human operator) may generate location data including a 10 mile radius value for the three WENDY'S® restaurants in San Francisco based on the population of the area including the WENDY'S® restaurants. The record generation module 110 may generate location data including a 100 mile radius for the WENDY'S® in Rock Springs.

[0086] In some implementations, the record generation module 110 may generate the location data 304 based on the number of similar places in the area in which the place described by the application state information 306 is located. For example, if a chain store (e.g., retail outlet or restaurant) has multiple different stores near one another, the record generation module 110 may generate the location data associated with a single store based on whether other similar stores are located nearby and/or how many stores are located nearby. In some examples, the record generation module 110 may generate location data 304 that defines a smaller geographic area tier places that are located near other similar places. For example, the record generation module 110 may generate a smaller radius value 326 in the location data 322 when the place described by the application state information 328 is located near other similar places. Shrinking the size of the geographic area defined in the location data 322 based on the presence (or number) of places nearby may reflect the notion that each of the similar places may have a more local area of interest to users. For example, users that drink coffee from stores in the Starbucks® chain may be interested more in locally relevant Starbucks stores, rather than those Starbucks stores that are located farther away.

[0087] In some implementations, the record generation module 110 may generate location data 304 based on the proximity of the place to transportation infrastructure, such as one or more highways. For example, the record generation module 110 may generate a larger geographic area for places that are in proximity to interstate highways. In some examples, the record generation module 110 may generate a geographic area having a shape that runs along the transportation infrastructure that is located in proximity to the place.

[0088] The record generation module 110 may also generate the location data 304 based on a variety of different parameters including, but not limited to, the uniqueness of the entity (e.g., restaurant), the relative density of a particular chain (e.g., store or restaurant), the type of the entity (e.g., movie theater, restaurant, department store) associated with the location, the uniqueness of the name of the entity, available delivery options associated with the entity, and user data associated with the entity (e.g., location of reviewers for a restaurant). For example, with respect to the uniqueness of the entity, more unique entities (e.g., those having a lower density of locations) may have a larger associated geographic area. With respect to the relative density of a chain, a smaller density of stores may cause each of the stores to have a larger associated geographic area. With respect to a uniqueness of the name of an entity, an entity having a more unique name a name including “airport” rather than “dry cleaner”) may be assigned a larger associated geographic area. With respect to available delivery options associated with the entity, the geographic area associated with the entity may be limited to the delivery range of the entity (e.g., the pizza delivery range of a pizza restaurant).

[0089] Although the record generation module 110 is described above as automatically generating and updating the location data 322 by automatically generating and updating the radius value 326, the record generation module 110 may automatically generate location data by generating other parameters of the location data in different examples. For example, in implementations where the location data defines shapes other than a circle, the record generation module 110 may generate other parameters in the location data that modifies the geographic area defined by the location data. In one example, if the location data 304 defines a rectangular shape, the record generation module 110 may modify the size of the rectangular shape by modifying the length and/or width of the rectangular shape. For example, the record generation module 110 may modify the length and/or width based on parameters including, but not limited to, the factors described above, such as the popularity of the place described in the application state information 306, population and/or population density, and the number of similar places in the area. In a similar manner, the record generation module 110 may generate and update the location data 304 by modifying different dimensional parameters for differently shaped geographic regions.

[0090] Some function records (e.g., function record 320) may include location data that defines a single geographic area (e.g., a single circular area). Other function records may include location data that defines a plurality of different geographic areas. For example, a function record may include multiple sets of location data that define multiple different geographic areas. In an example in which a function record includes location data that defines two different geographic areas, the function record may include a first set of location data that defines a first geographic area and a second set of

location data that describes a second geographic area. The multiple geographic areas of a function record may have basic shapes (e.g., circular or polygon) or more irregular shapes. The different geographic areas may be separated from one another. In other examples, the different geographic areas may overlap with one another. The location data defining a plurality of different geographic areas may be manually entered by a human operator. In other examples, the record generation module 110 may automatically generate the location data defining a plurality of different geographic areas.

[0091] A function record directed to multiple different places may include location data that defines multiple different geographic areas. In one example, a function record directed to a chain of stores (e.g., restaurants, convenience stores, gas stations, etc.) may include location data that defines a geographic area for each of the stores. The separate places may be located any distance from one another. For example, the places may be located in the same city, different states of the U.S., or different countries. In some examples, the different geographic areas may not overlap with one another. In other examples, some, or all of the geographic areas may overlap with one another.

[0092] Other function records directed to multiple different places may include function records associated with different concepts or categories, such as restaurant categories (e.g., breakfast restaurants or business categories (e.g., a gas station or a convenience store). For example, a function record associated with Chicago breakfast restaurants in the YELP® application may include location data for different breakfast restaurants in Chicago, Ill. The application state information may include data that describes breakfast restaurants in Chicago, the name of breakfast restaurants, along with other data that is relevant to breakfast entries in the YELP® application. In some examples, an access mechanism for the function record may include a link to an entry in the YELP® application that displays different breakfast restaurants in Chicago. In other examples, an access mechanism may include a link that performs a search for “Breakfast” in “Chicago, Ill.” in the YELP® application.

[0093] A function record that is associated with an application that offers service, or is otherwise relevant, in different geographic areas may include location data that defines a geographic area for each of the separate geographic areas where the application offers service. For example, if an application only offers service in distinct cities, then location data included a function record associated with the application may define multiple separate geographic areas corresponding to the distinct cities. Put another way, if an access mechanism of a function record references an application that provides service in distinct places, the location data included in the function record may define separate geographic areas corresponding to those distinct places.

[0094] FIG. 3D shows an example function record 330 associated with the OPENTABLE® application, developed by OpenTable, Inc. The OPENTABLE® application is a restaurant-reservation application that allows users to search for restaurants and make restaurant reservations. The OPENTABLE® application provides information about restaurants including descriptions of restaurants and user reviews of the restaurants. The example function record 330 of FIG. 3D describes an application state of the OPENTABLE® application in which the OPENTABLE® application accesses information for THE FRENCH LAUNDRY® restaurant.

[0095] The function record 330 includes the function ID “OPENTABLE—THE FRENCH LAUNDRY” indicated at 332, which may be used as a unique identifier to identify the function record 330. In other examples, the function ID 332 could include a URL as a unique identifier for the function record 330. For example, the function ID 332 may include the string “http://www.opentable.com/the-french-laundry” as a unique identifier for the function record 330. As described herein, such a function ID may be included in a web access mechanism of a function record. As another example, the function ID may have a different namespace than “http://,” such as “func://” in another example, the function ID could be a string of characters, numbers, and/or symbols that are not in human readable form.

[0096] The function record 330 includes application state information 334. The application state information 334 includes data fields for the restaurant category of THE FRENCH LAUNDRY® restaurant, a description of THE FRENCH LAUNDRY® restaurant, user reviews of THE FRENCH LAUNDRY® restaurant, and additional data fields. The restaurant category field may include the text “French cuisine” and “contemporary,” for example. The description field may include text that describes THE FRENCH LAUNDRY® restaurant. The user reviews field may include text of user reviews for THE FRENCH LAUNDRY® restaurant. The additional data fields may include additional data for THE FRENCH LAUNDRY® restaurant that may not specifically fit within the other defined fields, such as a menu for the restaurant, prices, and operating hours for the restaurant.

[0097] The function record 330 includes one or more access mechanism(s) 336. The access mechanism(s) 336 may include a reference to the OPENTABLE® application. An example application access mechanism for the function record 330 may include a reference to the OPENTABLE® native application along with one or more operations to be performed by the user device 102. For example, the application access mechanism may include an application resource identifier and/or one or more operations that cause the user device 102 to access the entry for THE FRENCH LAUNDRY® restaurant in the OPENTABLE® native application. An example application resource identifier may be “vnd.opentable.deeplink://opentable.com/restaurant/profile?rid=1180&refid=1.”

[0098] The function record 330 includes location data 338 that defines a geographic area associated with THE FRENCH LAUNDRY® restaurant. For example, the geographic area defined by the location data 338 may be a geographic area that includes THE FRENCH LAUNDRY® restaurant. In one example, the location data 338 may define a perimeter that surrounds a geographic area including THE FRENCH LAUNDRY® restaurant. The geographic area defined by the location data 338 may have a variety of different geometries. Since THE FRENCH LAUNDRY® restaurant is a world-renowned restaurant, the location data 338 may define a large geographic area around THE FRENCH LAUNDRY® restaurant. In an example where the location data 338 includes a coordinate and a radius value, the coordinate may include the geo-location of THE FRENCH LAUNDRY® restaurant. The radius value may range from a value of hundreds of miles up to worldwide coverage.

[0099] FIGS. 4A-4C show function records 400, 402, 404 that include a variety of different access mechanisms. The function record 400 of FIG. 4A includes a single application

access mechanism. The function record 402 of FIG. 4B includes multiple application access mechanisms. The function record 404 of FIG. 4C includes one or more application access mechanism(s), one or more web access mechanism(s), and one or more application download addresses.

[0100] Referring now to FIG. 4A, the function record 400 includes a function ID 406, application state information 408, and location data 410, as described above. The function record 400 also includes an application access mechanism 412. Accordingly, the function record 400 may include data related to a function of a native application and/or the state of the native application resulting from performance of the function. The data store 112 may include a plurality of function records having a similar structure as the function record 400.

[0101] The application access mechanism 412 may include a native application resource identifier 414 (i.e., an “application resource identifier”) and/or one or more operations 416 for a user device 102 to perform. For example, the application resource identifier 414 may be a string having an application specific scheme. The application resource identifier may include a reference to a native application and indicate one or more operations for the user device 102 (e.g., the native application) to perform. For example, the application resource identifier may include a reference to a native application, a domain name, and a path to be used by the native application to retrieve and display information to the user.

[0102] An example application resource identifier for the OPENTABLE® native application on the ANDROID® operating system is “vnd.opentable.deeplink://opentable.com/restaurant/profile?rid=88333&refid=1.” A portion of the example application resource identifier references the OPENTABLE® native application. For example, the substring “vnd.opentable.deeplink” of the application resource identifier references the OPENTABLE® native application. The example application resource identifier also indicates one or more operations for the OPENTABLE® native application to perform. For example, the OPENTABLE® native application may retrieve and display the information included in the application resource identifier domain and path defined by the substring “opentable.com/restaurant/profile?rid=88333&refid=1.” The user device 102 may launch the OPENTABLE® native application and display information retrieved from the location indicated in the application resource identifier in response to selection of a displayed search result including the example application resource identifier. The application resource identifier (e.g., the format) may be provided by the developer of the application in some examples.

[0103] In some examples, the application access mechanism 412 may include operations 416 for the user device 102 to perform in addition to the operation(s) indicated in the application resource identifier 414. For example, the search application 216, the operating system 224, and/or one of native applications 204 installed on the user device 102 may perform the operations 416 included in the application access mechanism 412 in order to set the native application into an application state specified by the application access mechanism 412. In some examples, the operations 416 may be included in a script. Examples of operations 416 may include, but are not limited to, launching a native application, waiting for the native application to start, creating and sending a search request to a server, setting a current geo-location in a native application, making a restaurant reservation, sending a text message, and adding an appointment to a calendar.

[0104] In some examples, the application access mechanism 412 may not include the application resource identifier 414 illustrated in FIG. 4A. Instead, the application access mechanism 412 can include operations 416 that reference a native application. The operations 416 may be performed by the user device 102. The one or more operations 416 may include instructions for at least one of the search application 216, the operating system 224, and one of native applications 204 on the user device 102. In response to selection of the application access mechanism 412 by the user, the user device 102 may perform the operations 416 included in the application access mechanism 412. In some examples, the operations 416 may be included in a script.

[0105] The application access mechanism 412 may also include edition information 418 that indicates the application edition with which the application access mechanism 412 is compatible. For example, the edition information 418 may indicate the operating system with which the application access mechanism 412 is compatible. In some examples, the search module 108 can determine whether to transmit the application access mechanism 412 in the search results 206 based on whether the user device 102 (e.g., the operating system 224) can handle the application access mechanism 412.

[0106] In some examples, an application resource identifier 414 is an application specific resource identifier that is defined by the developer of the application. In this example, the search application 216 receives the application resource identifier 414 and the operating system 224 may send the application resource identifier 414 to the native application referenced in the application resource identifier 414. The native application referenced in the application resource identifier 414 launches and is set into the state specified by the application resource identifier 414.

[0107] In some examples, an application function may not be accessible using an application resource identifier. For example, a function of the application may not include a corresponding application resource identifier that the native application may use to perform the function. As another example, some native applications may not be configured to receive an application resource identifier. In these examples, an application access mechanism for the native application can include one or more operations that cause the native application to perform the function that may not otherwise be accessible using an application resource identifier. For example, the search application 216 may receive the one or more operations (e.g., operations 416) and execute the one or more operations to set the native application into the desired application state. In a specific example, the one or more operations may include launching the native application along with additional operations for the native application to perform. For example, the search application 216 may initially trigger the native application to start and then wait for a period of time for the native application to start. Then, the search application 216 may perform additional operations included in the received application access mechanism, such as issuing a search instruction to the native application.

[0108] In still other examples, a native application may be configured to directly receive the operations transmitted by the search module 108. In these examples, the user device 102 may be configured to launch the native application according to the application access mechanism and then the launched native application may directly perform the operations received from the search module 108.

[0109] A single native application can provide a variety of different functionalities. For example, a restaurant reservation application can access reviews for a variety of different restaurants and set up reservations at a variety of different restaurants. Similarly, a travel application can book hotels, book flights, and provide reviews for different travel destinations. The different functionalities associated with a single native application may be accessed using a plurality of different application access mechanisms. For example, with respect to a restaurant reservation application, the data store 112 may include function records having different application access mechanisms for accessing different restaurant reviews and setting up reservations. Similarly, the data store 112 may include function records having different application access mechanisms for booking hotels, booking flights, and accessing reviews for different travel destinations.

[0110] The application access mechanisms for a single native application may vary in complexity. In some examples, the application access mechanisms may cause a native application to launch and then perform additional operations after launching, as described above. In other examples, application access mechanisms may cause an application to launch into a default state (e.g., a default homepage) without performing additional operations. A function record including an application access mechanism that causes an application to launch into a default state may be thought of as an access mechanism that is related to the native application, but not any particular state which may be accessed by the application. A function record including such an application access mechanism may include application state information describing the native application, instead of any particular application state. For example, the application state information may include the name of the developer of the native application, the publisher of the native application, a category (e.g., genre) of the native application, a description of the native application (e.g., a developer's description), and the price of the native application. The application state information may also include security or privacy data about the native application, battery usage of the native application, and bandwidth usage of the native application. The application state information may also include application statistics. Application statistics may refer to numerical data related to a native application. For example, application statistics may include, but are not limited to, a number of downloads, a download rate (e.g., downloads per month), a number of ratings, and a number of reviews.

[0111] FIG. 4B shows an example function record 402 including multiple different access mechanisms. Specifically, the function record 402 includes a first application access mechanism 420a and a second application access mechanism 420b (collectively "application access mechanisms 420"). The function record 402 also includes a function ID 422, location data 424, and application state information 426, as described above.

[0112] The first and second application access mechanisms 420 may be associated with first and second editions of a native application (i.e., first and second application editions), respectively. In general, different application access mechanisms of a function record may be associated with different operating systems and/or different versions of a native application. For example, different editions of a native application may refer to editions of the same native application that run on different operating systems. For example, the YELP® native application for ANDROID® and the YELP® native application for IOS® are two different editions of the YELP® native

application. Different editions of a native application may also refer to different versions (e.g., version 1.0, 2.0, phone version, tablet version, etc.) of a native application for the same operating system. For example, the YELP® native application may have first and second versions for the ANDROID® operating system. The different application access mechanisms **420** are used to access the same, or similar, functionality of the native application. For example, the different application access mechanisms **420** may retrieve similar information, play the same song, play the same movie, etc.

[0113] The example function record **402** of FIG. 4B includes application access mechanisms **420** that are associated with different operating systems. First and second application access mechanisms **420** are associated with the ANDROID® and IOS® operating systems, respectively. Each of the application access mechanisms **420** may include application resource identifiers that are specific to the different operating systems. Additionally, the operations included in the application access mechanisms **420** may also be specific to the different operating systems. As described above, the application access mechanisms **420** may include edition information.

[0114] The different application access mechanisms **420** included in the function record **402** may cause the corresponding application editions to launch and perform similar functions so that the application editions are set into similar application states. For example, the different application access mechanisms **420** included in the function record **402** may cause the corresponding application editions to be set into the application state described by the application state information **426**. Accordingly, although the application resource identifiers and/or the operations included in the application access mechanisms **420** may be different, the different application access mechanisms **420** may cause the different application editions to be set into similar application states. In one example, if the different application access mechanisms **420** reference different editions of an internet music player application, the different application access mechanisms **420** may cause the different application editions to play the same song. In another example, if the different application access mechanisms **420** reference different editions of a restaurant reservation application, the different application access mechanisms **420** may cause the different application editions to retrieve reservation information for the same restaurant.

[0115] FIG. 4C shows an example function record **404** that includes one or more web access mechanisms **428** and one or more application download addresses **430** along with one or more application access mechanisms **432**, a function ID **434**, location data **436**, and application state information **438**. The one or more web access mechanism **428** may be used by a wide variety of user devices **102** running different operating systems. In some examples, the web access mechanisms **428** include web resource identifiers, such as a uniform resource locator (URL) (i.e., a web address) used with hypertext transfer protocol (HTTP).

[0116] The web access mechanism(s) **428** (e.g., URLs) may be used by the web browser application **212** to access a web resource (e.g., a page of a web application/website) that includes similar information and/or performs similar functions as would be performed by a native application that receives the application access mechanism(s) **432**. For example, the web access mechanism(s) **428** may direct the

web browser application **212** to a web version of the native application referenced in the application access mechanism(s) **432**. If the function record **404** is for a specific Mexican restaurant in the YELP® application, the application access mechanism(s) **432** may include a reference to the YELP® native application and one or more operations that access an entry for the specific Mexican restaurant in the YELP® native application. In this example, the web access mechanism(s) **428** may include a web address that the web browser application **212** may use to access the entry for the specific Mexican restaurant on the YELP® application (e.g., web page of a web application/website).

[0117] The one or more application download addresses **430** indicate where the native applications for the one or more application access mechanisms **432** may be downloaded. An application download address **430** can be used by a user device **102** to download a native application referenced in an application access mechanism **432** in the event that the native application is not installed on the user device **102**. In some examples, an application download address **430** may include a web address (e.g., a URL) at which the native application can be previewed and downloaded. For example, an application download address **430** may direct the web browser application **212** to a digital distribution platform that is configured to distribute native applications. If a user device **102** includes a native download application for accessing a digital distribution platform, the application download address **430** may direct the installed native download application to a site where the native application referenced in the application access mechanism **432** can be downloaded. In some implementations, the function record **404** may include an application download address **430** for each native application edition referenced in the application access mechanism(s) **432**.

[0118] FIG. 5 shows an example method **500** for performing a location-based search. The method **500** is described with respect to the user device **102**, the search module **108**, and the data store **112** as illustrated in FIG. 2. In block **502**, the search module **108** receives the query wrapper **200**. In block **504**, the search module **108** determines the geographic location (i.e., geo-location) of the user device **102** based on data included in the query wrapper **200**.

[0119] In some examples, the search module **108** may determine the geo-location of the user device **102** in terms of latitude and longitude values that indicate the latitude and longitude of the user device **102**. In some examples, the search module **108** may determine the geo-location of the user device **102** in terms of an address, such as a postal address (e.g., street address). The geo-location of the user device determined by the search module **108** may be a point location in some examples (e.g., a latitude/longitude or a postal address). In other examples, the geo-location may be a larger area, e.g., a small area surrounding the user device **102**, such as a city block. In some examples, the user device **102** may generate geo-location data **218** (e.g., latitude and longitude) that indicates the geo-location of the user device **102**. For example, the user device **102** may generate the query wrapper **200** including the geo-location data **218**. In other examples, the search module **108** may determine the geo-location of the user device **102** based on data (e.g., the address **220**) included in the query wrapper **200**. For example, the search module **108** may look up the location of the user device **102** using the IP address **220**. In one example, the search module **108** may communicate with a remote server that can provide geo-location data for the user device **102** based on the

IP address 220. In other examples, the search module 108 may determine the geo-location of the user device 102 based on a user profile e stored at the search system 100) that is associated with the user device 102 that includes a user-preferred location or a user-frequented location.

[0120] In block 506, the search module 108 identifies function records in the data store 112 based on the search query 202. For example, the search module 108 may identify function records in the data store 112 by detecting matches between terms of the search query 202 and terms included in the application state information of the function records. Some, or all, of the function records identified in block 506 may include location data that the search module 108 may use to filter and/or score the function records.

[0121] In block 508, the search module 108 selects a set of function records from those function records identified in block 508. The set of function records selected from the identified function records may be referred to as a “consideration set” of function records. As described herein, the search module 108 may score the consideration set of function records and include information from the consideration set of function records in the search results 206.

[0122] In some implementations, the search module 108 may select function records from those function records identified in block 506 based on the geo-location of the user device 102 and the location data included in the identified function records. For example, the search module 108 may select a function record for inclusion in the consideration set if the geo-location of the user device 102 is included within the geographic area defined by the location data of the function record. In an example where the geo-location of the user device 102 is a point location, such as a latitude and longitude, the search module 108 may include a function record in the consideration set if the point location defined by the latitude and longitude is included within the geographic area defined by the location data of the function record.

[0123] For each of the function records identified in block 508, the search module 108 may determine whether to include the function record in the consideration set based on the location data included in the function record and the geo-location of the user device. The search module 108 may include function records in the consideration set in which the geo-location of the user device 102 is located within the geographic area defined by the location data of the function record. The search module 108 may exclude function records from the consideration set if the geo-location of the user device 102 is located outside of the geographic area defined by the location data of the function record. Put another way, the search module 108 may filter out those function records in which the location of the user device 102 is outside the geographic area defined by the location data of the function record. Filtering out function records in the manner described above may improve the relevancy of the search results 206 by limiting some search results to places that are geographically relevant to the user at search-time.

[0124] In block 510, the search module 108 scores the consideration set of function records. For example, the search module 108 may generate a score (e.g., a result score 226) for each of the function records that indicates the relevance of the function record to the search query 202. In some examples, the search module 108 may score a function record based on the location of the user device 102 within the geographic area defined by the location data of the function record. For example, the search module 108 may generate a larger result

score for function records associated with places that are closer to the user device 102 (i.e., the user). In some implementations, the search module 108 may score a function record based on how far outside of the perimeter the search location is located. For example, the search module 108 may generate reduced result scores for search locations located farther from the perimeter. In some examples, the search module 108 may determine the result score based on how far outside of the perimeter the search location is located relative to the size of the perimeter. For example, the search module 108 may tend to generate smaller result scores for function records if the search location is located farther away from a relatively small perimeter. In this example, the search module 108 may tend to generate larger result scores for function records if the search location is located closer to a relatively large perimeter.

[0125] In block 512, the search module 108 selects one or more access mechanisms from the function records for transmission in the search results 206. For example, the search module 108 may select access mechanisms from the function records associated with the largest result scores determined in block 510. In block 512, the search module 108 generates search results 206, and in block 514, the search module 108 transmits the search results 206 including a list of the selected application access mechanisms 208. In some implementations, the search module 108 may transmit all of the access mechanisms from the selected function records to the user device 102. In other implementations, the search module 108 may determine which access mechanisms are compatible with the user device 102 based on the platform data 222. In these implementations, the search module 108 may transmit a subset of the access mechanisms from the selected function records which are compatible with the user device 102 (e.g., based on OS version, web browser version, and/or device type).

[0126] The method 500 is an example method for performing a location-based search. In some implementations, the search module 108 may implement a location-based search in a manner that is different than that described in the method 500. In some implementations, the search module 108 may filter out function records based on location data, as described above with respect to block 508. In other implementations, the search module 108 may perform a location-based search by using the geo-location of the user device 102 to score function records, without performing a filtering operation described above in block 508. For example, the search module 108 may score a function record based on the location of the user device 102 within the geographic area defined by the location data. In still other implementations, the search module 108 may perform a location-based search by using the geo-location of the user device 102 to filter function records, but not use the geo-location data to score the function records (e.g., in block 510).

[0127] FIG. 6 shows another method 600 for performing a location-based search. In the method 600, the search module 108 may use a query-specified location or the geo-location of the user device 102 to filter and/or score the function records. The location used to filter and/or score the function records may be referred to as the “search location.” In the method 600, the search module 108 may use the geo-location of the user device 102 to filter and/or score function records in scenarios where the user does not specify a location in the search query 202. Alternatively, the search module 108 may use a query-

specified location included in the search query 202 instead of the geo-location of the user device 102 in order to filter and/or score function records.

[0128] In block 602, the search module 108 receives the query wrapper 200. In block 604, the search module 108 determines whether the search query 202 specifies a location, such as a postal address, zip code, street name, and/or a city name. If the search module 108 determines that the search query 202 specifies a location, the search module 108 may use the query-specified location as the search location, as indicated in block 606. In some examples, the search module 108 may map the query-specified location to a point location (e.g., a latitude and longitude). If the search module 108 determines that the search query 202 does not specify a location, the search module 108 may determine the geo-location of the user device 102 in block 608 and use the determined geo-location of the user device 102 as the search location, as indicated in block 610.

[0129] In block 612, the search module 108 identifies function records based on the search query 202. In block 614, the search module 108 selects a consideration set of function records based on the search location, as determined in either block 606 or block 610. As described above, the search module 108 may include a function record in the consideration set if the search location is located within the geographic area defined by the location data of the function record. If a function record includes location data that defines multiple different geographic areas, the search module 108 may include the function record in the consideration set if the search location is located within at least one of the multiple different geographic areas.

[0130] In block 616, the search module 108 scores the function records of the consideration set. In block 618, the search module 108 selects access mechanisms from the selected function records. In block 620, the search module 108 transmits the search results 206 to the user device 102.

[0131] FIG. 7 shows an example method 700 describing operation of an example user device 102. It may be assumed that the user device 102 described according to the method 700 includes a search application 216 (e.g., a native application or web browser implementation) that is configured to communicate with the search system 100.

[0132] In block 702, the search application 216 receives a search query from the user. In block 704, the user device 102 generates and transmits the query wrapper 200 to the search system 100. In block 706, the user device 102 waits for the search results 206 to be received. The method 700 continues in block 708 when the user device 102 receives the search results 206 from the search system 100. The search results 206 may include a list of access mechanisms 208. The search results 206 may also include result scores 226 associated with the access mechanisms 208. Additionally, the search results 206 may include link data 230 (e.g., text and/or images) for the access mechanisms 208. The search application 216 may generate user selectable links in the GUI based on the received link data 230.

[0133] In block 708, the search application 216 generates (e.g., renders) user selectable links based on the search results 206. In block 710, the search application 216 waits for the user to select one of the user selectable links. The method 700 continues in block 712 when the user selects (e.g., touches) one of the links. In response to selection of a link including an access mechanism, the user device 102 launches the applica-

tion referenced in the access mechanism and performs one or more operations indicated in the access mechanism in block 712.

[0134] FIG. 8 shows an example search module 108 that includes a query analysis module 800, a consideration set generation module 802 (hereinafter “set generation module 802”), and a consideration set processing module 804 (hereinafter “set processing module 804”). The query analysis module 800 receives the query wrapper 200. The query analysis module 800 analyzes the received search query 202. For example, the query analysis module 800 may perform various analysis operations on the received search query 202. Example analysis operations may include, but are not limited to, tokenization of the search query 202, filtering of the search query 202, stemming, synchronization, and stop word removal. In some implementations, the query analysis module 800 may detect a query-specified location included in the search query 202. For example, the query analysis module 800 may detect at least one of a street name, a street address, a city, a state, a zip code, or other location specified in the search query 202.

[0135] The set generation module 802 identifies a plurality of function records based on the received search query 202. In some examples, the set generation module 802 may identify the function records based on matches between terms of the search query 202 and terms in the function records. For example, the set generation module 802 may identify the function records based on matches between tokens generated by the query analysis module 800 and words included in the function records, such as words included in the application state information and/or function IDs.

[0136] The set generation module 802 may then determine a search location to use for generation of a “consideration set” of function records. The consideration set of function records may refer to the function records that are scored by the set processing module 804. The set generation module 802 may determine the geo-location of the user device 102 based on data included in the query wrapper 200, as described herein. In some examples, the set generation module 802 may use the geo-location of the user device 102 as the search location. In other scenarios, if the query analysis module 800 detects a query-specified location, the set generation module 802 may use the query-specified location as the search location.

[0137] The set generation module 802 may filter the identified function records based on the determined search location. For example, the set generation module 802 may include function records in the consideration set if the function records include location data that define a geographic area that includes the search location. In a more specific example, for each of the identified function records, the set generation module 802 may include the function record in the consideration set when the location data of the function record defines a geographic area that includes the search location. The set generation module 802 may exclude the function record from the consideration set if the search location is outside of the geographic area defined by the location data. Accordingly, the function records included in the consideration set may be geographically relevant to the user.

[0138] The set processing module 804 may score the function records in the consideration set in order to generate a set of search results 206. The scores associated with the function records may be referred to as “result scores.” The set processing module 804 may determine a result score for each of the function records in the consideration set. The result scores

associated with a function record may indicate the relative rank of the function record (e.g., the access mechanisms) among other function records. For example, a larger result score may indicate that a function record is more relevant to the received search query **202**.

[0139] The information conveyed by the search results **206** may depend on how the result scores **226** are calculated by the set processing module **804**. For example, the result scores **226** may indicate the relevance of an application function or application state to the search query **202**, the popularity of an application function or state, or other properties of the application function or state, depending on what parameters the set processing module **804** uses to score the function records.

[0140] The set processing module **804** may generate result scores for function records in a variety of different ways. In some implementations, the set processing module **804** generates a result score for a function record based on one or more scoring features. The scoring features may be associated with the function record and/or the search query **202**. A function record scoring feature (hereinafter “record scoring feature”) may be based on any data associated with a function record. For example, record scoring features may be based on any data included in the application state information of the function record. Example record scoring features may be based on metrics associated with a person, place, or thing described in the function record. Example metrics may include the popularity of a place described in the function record and/or ratings (e.g., user ratings) of the place described in the function record. In one example, if the function record describes a song, a metric may be based on the popularity of the song described in the function record and/or ratings (e.g., user ratings) of the song described in the function record. The record scoring features may also be based on measurements associated with the function record, such as how often the function record is retrieved during a search and how often access mechanisms of the function record are selected by a user. Record scoring features may also be based on whether the function record includes an application access mechanism that leads to a default state or a deeper native application state.

[0141] A query scoring feature may include any data associated with the search query **202**. For example, query scoring features may include, but are not limited to, a number of words in the search query **202**, the popularity of the search query **202**, and the expected frequency of the words in the search query **202**. A record-query scoring feature may include any data generated based on data associated with both the function record and the search query **202** that resulted in identification of the function record by the set generation module **802**. For example, record-query scoring features may include, but are not limited to, parameters that indicate how well the terms of the search query **202** match the terms of the application state information of the identified function record. The set processing module **804** may generate a result score for a function record based on at least one of the record scoring features, the query scoring features, and the record-query scoring features.

[0142] The set processing module **804** may determine a result score based on one or more of the scoring features listed herein and/or additional scoring features not explicitly listed. In some examples, the set processing module **804** may include one or more machine learned models (e.g., a supervised learning model) configured to receive one or more scoring features. The one or more machine learned models may generate result scores based on at least one of the record

scoring features, the query scoring features, and the record-query scoring features. For example, the set processing module **804** may pair the search query **202** with each function record and calculate a vector of features for each (query, record) pair. The vector of features may include one or more record scoring features, one or more query scoring features, and one or more record-query scoring features. The set processing module **804** may then input the vector of features into a machine-learned regression model to calculate a result score for the function record. In some examples, the machine-learned regression model may include a set of decision trees (e.g., gradient boosted decision trees). In another example, the machine-learned regression model may include a logistic probability formula. In some examples, the machine learned task can be framed as a semi-supervised learning task, where a minority of the training data is labeled with human curated scores and the rest are used without human labels.

[0143] The result scores **226** associated with the function records (e.g., access mechanisms) may be used in a variety of different ways. The set processing module **804** and/or the user device **102** may rank the access mechanisms **208** based on the result scores **226** associated with the access mechanisms **208**. In these examples, a larger result score may indicate that the access mechanism (e.g., the function or application state) is more relevant to a user than an access mechanism having a smaller result score. In examples where the user device **102** displays the search results **206** as a list, the user device **102** may display the links for access mechanisms having larger result scores nearer to the top of the results list (e.g., near to the top of the screen) in these examples, the user device **102** may display the links for access mechanisms having lower result scores farther down the list (e.g., off screen).

[0144] In some implementations, the set processing module **804** may score function records based on where the search location is located within the geographic area defined by the location data of the function record. For example, the set processing module **804** may assign a larger result score to a function record in circumstances when the search location is nearer to the place described by the function record. With respect to a function record that defines a coordinate and a perimeter (e.g., a circle), the set processing module **804** may assign a larger score to a function record when the search location is closer to the coordinate. The set processing module **804** may assign a smaller result score to a function record when the search location is farther from the coordinate (e.g., closer to the perimeter of the geographic area). In one example, the set processing module **804** may generate a scoring feature indicating the distance from the search location to the coordinate. For example, the scoring feature may be a number between 0.0 and 1.0 that indicates the distance of the search location from the coordinate. In this example, a 0.0 may correspond to a search location that is located at the coordinate (e.g., coordinates **314**, **324**) of the geographic area (e.g., circular area). A 1.0 may correspond to a search location that is located at the perimeter of a geographic area (e.g., the perimeter of a circular geographic area).

[0145] As described herein, in some implementations, the search module **108** may perform a location-based search by using the search location to score function records without performing a filtering operation described above. For example, the set processing module **804** may score the function records of the consideration set based on whether the search location is included in the geographic region defined by the location data. In a more specific example, the set

processing module **804** may generate a scoring feature of 0 or 1 that indicates whether the search location is included in the geographic region. The set processing module **804** may then use the generated scoring feature to score the function record. The set processing module **804** may assign a larger result score to a function record when the search location is included in the geographic area. The set processing module **804** may assign a smaller score if the search location is outside of the geographic area. In some implementations, the set processing module **804** may score a function record based on how far outside of the perimeter the search location is located. For example, the set processing module **804** may assign a smaller score when the search location is located farther outside of the geographic area. The set processing module **804** may also determine the result score based on how far outside of the perimeter the search location is located relative to the size of the perimeter.

[0146] As described above, the set processing module **804** may score a function record based on a variety of different factors. In some examples, the set generation module **802** may perform similar scoring operations on the function records prior to the set processing module **804**. For example, the set generation module **802** may score a function record based on text matches with the search query **202**, the search location, and location data of the function record. In some examples, the set generation module **802** may exclude function records from the consideration set based on the scores associated with the function records. For example, the set generation module **802** may select the highest scoring function records for inclusion into the consideration set.

[0147] Although function records in the data store **112** may include location data, some function records in the data store **112** may not include location data. For example, a function record may not include location data when the function record is associated with a function that is not specific to a geographic area. The search module **108** of the present disclosure may be configured to return search results that include access mechanisms which are not associated with location data.

[0148] FIG. 9 shows an example method **900** for performing a location-based search. The method **900** is described with reference to the search module **108** of FIG. 8. In block **902**, the query analysis module **800** receives a query wrapper **200**. In block **904**, the query analysis module **800** analyzes the search query **202**. In some implementations, the query analysis module **800** may determine whether the search query **202** specifies a location, such as a street address, zip code, or city name.

[0149] In block **906**, the set generation module **802** determines the search location to be used during the search for function records. If the query analysis module **800** determines that the search query **202** includes a location, then the set generation module **802** may set the detected location (i.e., the query-specified location) as the search location. If the query analysis module **800** determines that the search query **202** does not include a location, the set generation module **802** may set the search location to the location of the user device **102**, as determined based on other data included in the query wrapper **200**, such as geo-location data indicating latitude and longitude of the user device **102**.

[0150] In block **908**, the set generation module **802** identifies a consideration set of function records based on the search query **202** and the search location, as described above. In block **910**, the set processing module **804** scores the function

records of the consideration set. In some examples, the set processing module **804** may score the function records based on the search location. In block **912**, the set processing module **804** selects access mechanisms from the scored function records. For example, the set processing module **804** may select access mechanisms from the function records associated with the largest result scores. In block **914**, the set processing module **804** transmits search results **206** to the user device **102**.

[0151] Although the search system **100** may be configured to generate location-based search results including a variety of different access mechanisms described above, in some examples, the search system **100** may be configured to provide location-based search results that are more directed to assisting a user in finding native applications to download. For example, the search system **100** may be tailored to providing the user with links including application download addresses in response to a query wrapper **200** so that a user may find native applications to install.

[0152] FIGS. 10-11 illustrate an example method and user interface that are directed to assisting a user in finding native applications to download. With respect to FIGS. 10-11, it may be assumed that the search system **100** is configured to receive query wrappers and find native applications for download from a digital distribution platform. It may also be assumed that the search system **100** is configured to generate a list of application download addresses in response to a received query wrapper. The user device **102** may generate download links including the application download addresses.

[0153] The search system **100** may be configured in a variety of different ways in order to provide the features that are illustrated and described with respect to FIGS. 10-11. In some implementations, the search system **100** may operate as described above, but instead of transmitting any combination of access mechanisms, the search system **100** may be configured to transmit application download addresses (e.g., only application download addresses).

[0154] In other implementations, the search system **100** may be configured to search for those function records related to native applications, but not any particular state which may be accessed by the native application, as described above. The search system **100** may also be configured to return search results to the user associated with the function records, such as application download addresses for the native application described in the function records. As further described above, these function records may include application state information describing the native application, instead of a particular state. Example function records of this type may include information, such as the name of the developer of the native application, the publisher of the native application, a category (e.g., genre) of the native application, a description of the native application (e.g., a developer's description), application statistics, the price of the native application, security or privacy data about the native application, battery usage of the native application, and bandwidth usage of the native application.

[0155] A function record related to a native application, but not any particular state of the native application, may include location data that defines one or more geographic areas in which the native application is relevant to the user. For example, some native applications may only provide service in certain areas (e.g., a certain city). In a more specific example, a taxi reservation application may only provide

service in the one or more cities in which the taxi company operates. In this specific example, the location data of the function record may include location data that defines the one or more geographic regions in which the taxi company operates (i.e., the areas in which the taxi reservation application is relevant).

[0156] In some examples, some native applications may have restrictions on location, such as a media streaming application that is allowed only in certain countries, or a gambling application that is not allowed to be used outside of a specific state of the United States. In these examples, the location data may define geographic areas in which the native application is not restricted (i.e., the non-restricted areas). In some examples, some native applications may have a primary area of usefulness or interest based on the regions served by the application. Some examples of native applications associated with a primary area of usefulness/interest may include, but are not limited to, local transportation native applications, television network related applications such as television scheduling/streaming apps, native applications for local sporting teams, and native applications for local events, such as a fair or concert. In these examples, the location data may define the geographic area of usefulness/interest.

[0157] FIG. 10 is a flow diagram illustrating an example method 1000 for performing a location-based search directed to assisting a user in finding native applications to download. In block 1002, the search module 108 receives the query wrapper 200 from the user device 102 and analyzes the search query 202. In block 1004, the search module 108 determines the search location to use during the search, as described above.

[0158] In block 1006, the search module 108 generates a consideration set of function records related to native applications based on the determined search location. For example, each of the function records in the consideration set may include application state information that describes a native application. Each of the function records in the consideration may describe a different native application. Accordingly, each of the function records may include application download addresses for different native applications.

[0159] In block 1008, the search module 108 may score the function records of the consideration set based on the search location. In block 1010, the search module 108 may select application download addresses from the scored function records. In block 1012, the search module 108 transmits search results to the user device 102. The search results may include application download addresses and link data for generating user selectable links including the application download addresses.

[0160] An example location-based search directed to assisting a user in finding native applications to download is now described. In this example, it may be assumed that there are three function records directed to public transit native applications in the data store 112. The three native applications may include: 1) a Seoul Korea Subway application that lists transit routes in Seoul, South Korea, 2) a San Francisco Train application that lists transit routes in San Francisco, Calif., and 3) a New York City Subway application that lists subway routes in New York City, N.Y. Each of the three native applications may be relevant to users in their respective cities and surrounding areas. Accordingly, for each of the three function records, the record generation module 110 (or a human operator) may generate location data that includes a geographic area surrounding the respective city. For example,

the record generation module 110 may generate location data that includes a coordinate at the city (e.g., the center of the city) and a radius value of 100-200 miles.

[0161] In this example, a user near New York City searching for “subway” may be returned search results including the New York City Subway application, but not returned the San Francisco Train application or the Seoul Korea Subway application. A user located in San Francisco searching for “Public Transit” may be returned search results including the San Francisco Train application, but not the New York City Subway application or the Seoul Korea Subway application. A user located anywhere in the world searching for “Seoul Subway” may be returned search results including the Seoul Korea Subway application because the search module 108 may set the search location to Seoul Korea based on the determination that the query specified the location “Seoul.”

[0162] FIG. 11 shows an example GUI for a search application directed to assisting a user in finding native applications to download. In FIG. 11, the user entered a search query “Late night food.” In this example, the search module 108 returned application download addresses for the YELP® native application, the DOMINO’S PIZZA® native application, the TRIPADVISOR® native application, the OPENTABLE® native application, and the URBANSPOON® native application. The application download addresses are included in the user selectable links 1100a, 1100b, 1100c, 1100d, and 1100e illustrated in FIG. 11.

[0163] Modules and data stores included in the search system 100 represent features that may be included in the search system 100 of the present disclosure. For example, the search module 108, the function record generation/update module 110, and the data store 112 may represent features included in the search system 100. The modules and data stores described herein may be embodied by electronic hardware, software, firmware, or any combination thereof. Depiction of different features as separate modules and data stores does not necessarily imply whether the modules and data stores are embodied by common or separate electronic hardware or software components. In some implementations, the features associated with the one or more modules and data stores depicted herein may be realized by common electronic hardware and software components. In some implementations, the features associated with the one or more modules and data stores depicted herein may be realized by separate electronic hardware and software components.

[0164] The modules and data stores may be embodied by electronic hardware and software components including, but not limited to, one or more processing units, one or more memory components, one or more input/output (I/O) components, and interconnect components. Interconnect components may be configured to provide communication between the one or more processing units, the one or more memory components, and the one or more I/O components. For example, the interconnect components may include one or more buses that are configured to transfer data between electronic components. The interconnect components may also include control circuits (e.g., a memory controller and/or an I/O controller) that are configured to control communication between electronic components.

[0165] The one or more processing units may include one or more central processing units (CPUs), graphics processing units (GPUs), digital signal processing units (DSPs), or other processing units. The one or more processing units may be configured to communicate with memory components and

I/O components. For example, the one or more processing units may be configured to communicate with memory components and I/O components via the interconnect components.

[0166] A memory component may include any volatile or non-volatile media. For example, memory may include, but is not limited to, electrical media, magnetic media, and/or optical media, such as a random access memory (RAM), read-only memory (ROM), non-volatile RAM (NVRAM), electrically-erasable programmable ROM (EEPROM), Flash memory, hard disk drives (HDD), magnetic tape drives, optical storage technology (e.g., compact disc, digital versatile disc, and/or Blu-ray Disc), or any other memory components.

[0167] Memory components may include (e.g., store) data described herein. For example, the memory components may include the data included in the function records of the data store **112**. Memory components may also include instructions that may be executed by one or more processing units. For example, memory may include computer-readable instructions that, when executed by one or more processing units, cause the one or more processing units to perform the various functions attributed to the modules and data stores described herein.

[0168] The I/O components may refer to electronic hardware and software that provides communication with a variety of different devices. For example, the I/O components may provide communication between other devices and the one or more processing units and memory components. In some examples, the I/O components may be configured to communicate with a computer network. For example, the I/O components may be configured to exchange data over a computer network using a variety of different physical connections, wireless connections, and protocols. The I/O components may include, but are not limited to, network interface components (e.g., a network interface controller), repeaters, network bridges, network switches, routers, and firewalls. In some examples, the I/O components may include hardware and software that is configured to communicate with various human interface devices, including, but not limited to, display screens, keyboards, pointer devices (e.g., a mouse), touchscreens, speakers, and microphones. In some examples, the I/O components may include hardware and software that is configured to communicate with additional devices, such as external memory (e.g., external HDDs).

[0169] In some implementations, the search system **100** may be a system of one or more computing devices (e.g., a computer search system) that are configured to implement the techniques described herein. Put another way, the features attributed to the modules and data stores described herein may be implemented by one or more computing devices. Each of the one or more computing devices may include any combination of electronic hardware, software, and/or firmware described above. For example, each of the one or more computing devices may include any combination of processing units, memory components, I/O components, and interconnect components described above. The one or more computing devices of the search system **100** may also include various human interface devices, including, but not limited to, display screens, keyboards, pointing devices (e.g., a mouse), touchscreens, speakers, and microphones. The computing devices may also be configured to communicate with additional devices, such as external memory (e.g., external HDDs).

[0170] The one or more computing devices of the search system **100** may be configured to communicate with the network **106**. The one or more computing devices of the search system **100** may also be configured to communicate with one another (e.g., via a computer network). In some examples, the one or more computing devices of the search system **100** may include one or more server computing devices configured to communicate with user devices (e.g., receive query wrappers and transmit search results), gather data from data sources **104**, index data, store the data, and store other documents. The one or more computing devices may reside within a single machine at a single geographic location in some examples. In other examples, the one or more computing devices may reside within multiple machines at a single geographic location. In still other examples, the one or more computing devices of the search system **100** may be distributed across a number of geographic locations.

What is claimed is:

1. A method comprising:

receiving, at a computing device, a search query from a user device;

identifying, by the computing device, a plurality of function records included in a data store based on the received search query, each of the function records comprising:

an access mechanism specifying a state of an application;

state information corresponding to the state of the application specified by the access mechanism; and

location data including a coordinate and a perimeter, wherein the coordinate defines the location of a place corresponding to the state information and the perimeter defines a geographic area surrounding the coordinate;

determining, by the computing device, a search location;

for each of the identified function records, determining a score tier the function record based on where the determined search location is located relative to the geographic area defined by the location data of the function record;

selecting, by the computing device, access mechanisms from the function records based on the scores associated with the function records; and

transmitting the selected access mechanisms from the computing device to the user device.

2. The method of claim 1, wherein the score indicates a relevance of the function record to the search query and the search location.

3. The method of claim 1, wherein determining the score for the function record comprises determining the score based on whether the search location is located within the geographic area defined by the perimeter.

4. The method of claim 3, wherein determining the score for the function record comprises:

determining a first score when the search location is located within the geographic area defined by the perimeter; and

determining a second score that is different than the first score when the search location is located outside of the geographic area defined by the perimeter.

5. The method of claim 3, wherein, when the search location is located within the perimeter, determining the score for the function record based on where the search location is located relative to the coordinate.

6. The method of claim 3, wherein, when the search location is located outside of the perimeter, determining the score for the function record based on the distance of the search location from the perimeter.

7. The method of claim 1, further comprising generating, by the computing device, the perimeter for at least one of the function records based on the population of the area surrounding the coordinate.

8. The method of claim 1, further comprising generating, by the computing device, the perimeter for at least one of the function records based on the population density of the area surrounding the coordinate.

9. The method of claim 1, further comprising generating, by the computing device, the perimeter for at least one of the function records based on the proximity of the coordinate to transportation infrastructure.

10. The method of claim 1, further comprising generating, by the computing device, the perimeter for at least one of the function records based on the popularity of the place defined by the state information.

11. The method of claim 1, wherein, for at least one of the plurality of function records, the perimeter is a circular perimeter defining a circular geographic area surrounding the coordinate, and wherein the coordinate defines a center point of the circular perimeter.

12. The method of claim 11, wherein the location data includes a radius value around the coordinate that defines the circular perimeter, and wherein the method further comprises generating the radius value based on the population of the geographic area surrounding the coordinate.

13. The method of claim 1, wherein, for at least one of the plurality of function records, the perimeter is a polygon perimeter defining a geographic area surrounding the coordinate, and wherein the coordinate defines a center point of the polygon perimeter.

14. The method of claim 1, further comprising, for each of the plurality of function records:

determining, by the computing device, whether the search location is located within the geographic area defined by the location data of the function record; and

filtering out, by the computing device, the function record if the determined search location is located outside of the geographic area defined by the location data of the function record, wherein determining the score for the function record comprises determining the score for the function record based on where the determined search location is located within the geographic area defined by the location data of the function record.

15. The method of claim 1, further comprising receiving, at the computing device, data from the user device indicating the geo-location of the user device, wherein determining the search location comprises determining the search location based on at least one of a location specified in the search query and the data from the user device indicating the geo-location of the user device.

16. A system comprising:

a non-transitory data store that includes function records, each of the function records comprising:
an access mechanism specifying a state of an application;
state information corresponding to the state of the application specified by the access mechanism; and
location data including a coordinate and a perimeter, wherein the coordinate defines the location of a place

corresponding to the state information and the perimeter defines a geographic area surrounding the coordinate; and

one or more computing devices in communication with the non-transitory data store and configured to:

receive a search query from a user device;

identify a plurality of function records included in the data store based on the received search query;

determine a search location;

for each of the identified function records, determine a score for the function record based on where the determined search location is located relative to the geographic area defined by the location data of the function record;

select access mechanisms from the function records based on the scores associated with the function records; and

transmit the selected access mechanisms to the user device.

17. The system of claim 16, wherein the one or more computing devices are configured to generate the perimeter for at least one of the function records based on the population of the area surrounding the coordinate.

18. The system of claim 16, wherein the one or more computing devices are configured to generate the perimeter for at least one of the function records based on the population density of the area surrounding the coordinate.

19. The system of claim 16, wherein, for at least one of the plurality of function records, the perimeter is a circular perimeter defining a circular geographic area surrounding the coordinate, and wherein the coordinate defines a center point of the circular perimeter.

20. The system of claim 19, wherein the location data includes a radius value around the coordinate that defines the circular perimeter, and wherein the one or more computing devices are configured to generate the radius value based on the population of the geographic area surrounding the coordinate.

21. The system of claim 16, wherein, for each of the plurality of function records, the one or more computing devices are configured to:

determine whether the search location is located within the geographic area defined by the location data of the function record; and

filter out the function record if the determined search location is located outside of the geographic area defined by the location data of the function record, wherein the one or more computing devices are configured to determine the score for the function record based on where the determined search location is located within the geographic area defined by the location data of the function record.

22. A non-transitory computer-readable storage medium comprising instructions that cause one or more computing devices to:

receive a search query from a user device;

identify a plurality of function records included in a data store based on the received search query, each of the function records including:

an access mechanism specifying a state of an application;

state information corresponding to the state of the application specified by the access mechanism; and

location data including a coordinate and a perimeter, wherein the coordinate defines the location of a place corresponding to the state information and the perimeter defines a geographic area surrounding the coordinate;

determine a search location;

for each of the identified function records, determine a score for the function record based on where the determined search location is located relative to the geographic area defined by the location data of the function record;

select access mechanisms from the function records based on the scores associated with the function records; and

transmit the selected access mechanisms to the user device.

* * * * *