(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 May 2018 (11.05.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/085482 A1

(51) International Patent Classification:
*G06Q 10/10* (2012.01) *G06F 17/30* (2006.01)

(21) International Application Number:
PCT/US2017/059636

(22) International Filing Date:
02 November 2017 (02.11.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/418,268 06 November 2016 (06.11.2016) US
15/450,758 06 March 2017 (06.03.2017) US

(71) Applicant: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **FOWLER, Chad**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US). **MATEEV, Benjamen Ljudmilov**; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) Agent: **MINHAS, Sandip S.** et al.; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ,

(54) Title: EFFICIENCY ENHANCEMENTS IN TASK MANAGEMENT APPLICATIONS

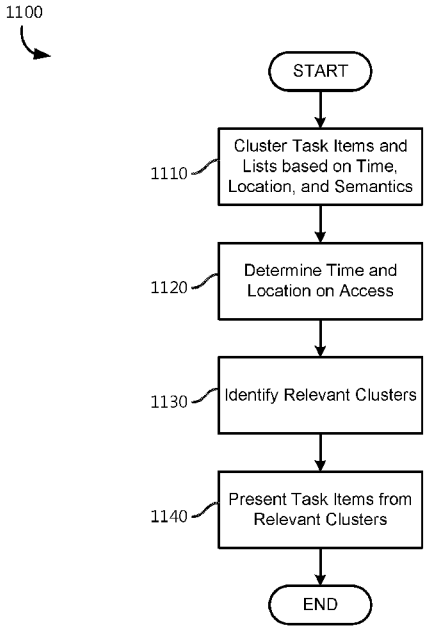



FIG. 11

(57) Abstract: Efficiency improvements for electronic task managers and an improved user experience are realized when more relevant and fewer irrelevant tasks are presented to users and users are given greater control in manipulating those task items. By heuristically determining times, locations, and semantics associated with task relevance and integrating the management of tasks into more applications, the functionality of the systems providing for electronic task management is improved, as computer resources are spent with greater utility to the users and the user experience is improved for the users.

WO 2018/085482 A1

[Continued on next page]

UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

**Declarations under Rule 4.17:**

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

**Published:**

— *with international search report (Art. 21(3))*

# EFFICIENCY ENHANCEMENTS IN TASK MANAGEMENT APPLICATIONS

## BACKGROUND

**[0001]** Electronic task management systems and applications enable users to track various tasks more efficiently than with hardcopy notes; users can access the same tasks from multiple devices, rearrange the tasks, and share tasks between users remotely. The ease of adding tasks to an electronic task manager, however, can leave users overwhelmed; too many, irrelevant, or contextually inappropriate tasks can distract the user from the tasks that are relevant to the user at a given time and place. The provision of unwanted tasks not only degrades the user experience, but also wastes computing resources that are used to provide tasks that are not wanted by the user that could be used more efficiently for other tasks.

## SUMMARY

**[0002]** This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description section. This summary is not intended to identify all key or essential features of the claimed subject matter, nor is it intended as an aid in determining the scope of the claimed subject matter.

**[0003]** Enhancements to the efficiency of a task management application are discussed herein in relation to systems, methods, and computer readable media that provide such enhancements. Relational data for entities and the context in which users interact with task items, including the productivity applications used to complete task items, are used to provide users with more relevant tasks, fewer irrelevant tasks, and with greater control and convenience in manipulating task items.

**[0004]** In one aspect, task items are clustered by their interaction contexts to provide more relevant results to users at different times and locations. Contextual information related to when and where tasks items are added to task lists, marked as complete, viewed, deferred, cancelled, etc., is used in conjunction with semantic data from the entities related to those task items to cluster various task items and task lists. Depending on the time and location at which the user accesses a task management application, a relevant cluster – having an equivalent or similar time and/or location to the user's time and/or location of access – is selected from which task items are to be presented. The user is thus presented with more task items with greater relevance to the time and location at which the task manager is accessed.

**[0005]** By providing enhanced efficiency for a task management application, not only is the user's experience improved, but the functionality of the device used to provide the task management application is also improved. The device spends computing resources (processor cycles and memory storage space) with greater precision; wasting fewer resources to provide unwanted tasks for the user's consideration.

**[0006]** Examples are implemented as a computer process, a computing system, or as an article of manufacture such as a device, computer program product, or computer readable medium. According to an aspect, the computer program product is a computer storage medium readable by a computer system and encoding a computer program comprising instructions for executing a computer process.

**[0007]** The details of one or more aspects are set forth in the accompanying drawings and description below. Other features and advantages will be apparent from a reading of the following detailed description and a review of the associated drawings. It is to be understood that the following detailed description is explanatory only and is not restrictive of the claims.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0008]** The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate various aspects. In the drawings:

**FIGURE 1** illustrates an example operating environment in which the present disclosure may be practiced;

**FIGURES 2A** and **2B** are illustrations of example graphical user interfaces in which a task list is implemented;

**FIGURE 3** illustrates an example graphical user interface in which is displayed suggested tasks on which a user may focus;

**FIGURE 4A** illustrates a calendar application showing one or more events illustrated in relation to a calendar and order of occurrence for a day;

**FIGURE 4B** illustrates a Plan of tasks for the user based on data received from the calendar application illustrated in **FIGURE 4A**;

**FIGURE 5** illustrates an example graphical user interface in which Task List implementation is provided;

**FIGURE 6** illustrates a graphical user interface in which a task application provides task list items in a task list category along with additional data about the selected task list item;

**FIGURE 7** illustrates a graphical user interface in which task list templates are provided to allow users to get specific information about a content item provided as a task item template, such as, for example, a job or a location;

**FIGURE 8** illustrates various source attributions for suggested tasks items;

**FIGURE 9** illustrates an example user interface for a digital assistant to create task items;

**FIGURES 10A** and **10B** illustrate example user interfaces for a time period view of a set of task items as the time period's task list is constructed;

**FIGURE 11** is a flow chart showing general stages involved in an example method for enhancing efficiency by clustering task items by their interaction contexts to provide more relevant results to users at different times and locations;

**FIGURE 12** is a block diagram illustrating example physical components of a computing device;

**FIGURES 13A** and **13B** are block diagrams of a mobile computing device; and

**FIGURE 14** is a block diagram of a distributed computing system.

## DETAILED DESCRIPTION

**[0009]** The following detailed description refers to the accompanying drawings. Wherever possible, the same reference numbers are used in the drawings and the following description refers to the same or similar elements. While examples may be described, modifications, adaptations, and other implementations are possible. For example, substitutions, additions, or modifications may be made to the elements illustrated in the drawings, and the methods described herein may be modified by substituting, reordering, or adding stages to the disclosed methods. Accordingly, the following detailed description is not limiting, but instead, the proper scope is defined by the appended claims. Examples may take the form of a hardware implementation, or an entirely software implementation, or an implementation combining software and hardware aspects. The following detailed description is, therefore, not to be taken in a limiting sense.

**[0010]** Enhancements to the efficiency of a task management application are discussed herein in relation to systems, methods, and computer readable media that provide such enhancements. Relational data for entities and the context in which users interact with task items, including the productivity applications used to complete task items, are used to provide users with more relevant tasks, fewer irrelevant tasks, and with greater control and convenience in manipulating task items.

**[0011]** In one aspect, task items are clustered by their interaction contexts to provide more relevant results to users at different times and locations. Contextual information related to when and where tasks items are added to task lists, marked as complete, viewed, deferred, cancelled, etc., is used in conjunction with semantic data from the entities related to those task items to cluster various task items and task lists. Depending on the time and location at which the user accesses a task management application, a relevant cluster – having an equivalent or similar time and/or location to the user's time and/or location of access – is selected from which task items are to be presented. The user is thus presented with more task items with greater relevance to the time and location at which the task manager is accessed.

**[0012]** By providing enhanced efficiency for a task management application, not only is the user's experience improved, but the functionality of the device used to provide the task management application is also improved. The device spends computing resources (processor cycles and memory storage space) with greater precision; wasting fewer resources to provide unwanted tasks for the user's consideration.

**[0013]** **FIGURE 1** illustrates an example operating environment **100** in which the present disclosure may be practiced. As illustrated, a user device **110** is in communication with a task list service **120**. In various aspects, the task list service **120** is hosted on the user device **110**, while in other aspects, the task list service **120** is hosted on a remote device as a service accessible by the user device **110**. The task list service **120** includes in various configurations one or more of a: heuristic engine **121**, a suggestion engine **122**, a context clusterer **123**, a preview generator **124**, user profiles **125**, a context listener **126**, and a relational store **127**. The task list service **120** is in further communication with one or more services that may be hosted on the user device **110** or another device that include, but are not limited to: a relational graph service **130**, an email service **140**, a calendar service **150**, and a productivity service **160**. Although only one of each component is illustrated in **FIGURE 1**, it will be appreciated that in different aspects more than one of one or more components are provided (e.g., more than one user device **110** is in communication with the task list service **120**, which is in communication with more than one email service **140**).

**[0014]** Each of the user device **110**, task list service **120**, and the services **130-160** are illustrative of a multitude of computing systems including, without limitation, desktop computer systems, wired and wireless computing systems, mobile computing systems (e.g., mobile telephones, netbooks, tablet or slate type computers, notebook computers,

and laptop computers), hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, printers, and mainframe computers. The hardware of these computing systems is discussed in greater detail in regard to **FIGURES 12**, **13A**, **13B**, and **14**. User devices **110** are operated by users, who may be humans or automated systems (e.g., "bots"). In various aspects, the user device **110**, task list service **120**, and the services **130-160** may be accessed by a user or each other locally and/or by a network, which may include the Internet, a Local Area Network (LAN), a private distributed network for an entity (e.g., a company, a university, a government agency), a wireless *ad hoc* network, a Virtual Private Network (VPN) or other direct data link (e.g., Bluetooth connection, a direct wired link).

**[0015]** The user device **110** is accessed by a user to operate a task list application, among other features and applications. The task list application provides user-specific tasks that the user wishes to be reminded of to complete and tools for manipulating those tasks (e.g., assign task to another user, share task with another user, complete task, mark status of task, add task, remove task). For example, a user may access the task list application to receive a reminder to pay rent on a given set of days, to attend a meeting at a given time, or to go grocery shopping at an undefined time. In various aspects, the task list application is provided by the task list service **120** in a thin client running on the user device **110** in conjunction with a client running on a remote server. In other aspects, the task list application is provided by a task list service **120** running on the user device **110** as a thick client. In yet other aspects, the task list service **120** operates as a distributed system, running on the user device **110** as a thick client when a network connection to the remote server is not available (or not needed) and as a thin client when the network connection is available.

**[0016]** The task list service **120** includes one or more components that may be enabled or disabled as users enable or disable features or network connections to a remote server are established or lost. In various aspects, a task list service **120** local to a given user device **110** may also disable or reduce in size or complexity one or more components compared to a task list service **120** that is accessible over a network by multiple user devices **110**.

**[0017]** A heuristic engine **121** is operable to learn user behavior over time to enhance the determinations of which candidate tasks discovered from task sources are to be presented, and in what order, to a given user at a given time and location. The heuristic

engine **121** is operable to use one or more machine learning approaches to determine how to best serve the needs and use-cases presented by individual users.

**[0018]** A suggestion engine **122** is operable to determine whether a candidate task received from a tasks source should be suggested to the user as a task to perform at a given time and/or location. From all of the candidate task items that may be presented to the user at any given time, the suggestion engine **122** filters those task items to a manageable subset based on the user's existing task items (to avoid scheduling conflicts), prior acceptances/rejections of suggested task items, and the prior actions of the user. For example, if a user's calendar includes an event for an upcoming birthday, a suggested task is created that the person whose birthday is coming up should be called prior to that date. In another example, where the user sent an email that included a promise to send an attachment by a deadline, a task is suggested to meet this deadline. In a further example, an important meeting is observed on the calendar service **150** as occurring on Friday, and the suggestion service **122** will observe the rest of the week's calendar to determine which days prior to the meeting are likely to allow for a task item to prepare for the important meeting. For example, the task item will be presented on Monday and Thursday, but not Tuesday or Wednesday, due to the number of task items already accepted for on those days (Tuesday and Wednesday being busier or having more task items accepted in the present example than Monday and Thursday).

**[0019]** A content clusterer **123** is operable to cluster tasks and entities that are related in the location and time of interaction, and the semantics of terms that they contain. As will be understood, clustering is a statistical operation that groups items based on shared characteristics (and combinations thereof). In one aspect, tasks interacted with (created/completed) with similar time ranges are clustered together based on similar time characteristics. In another aspect, tasks interacted with (created/completed) when the user is at a given location will be clustered together based on location characteristics. In a further aspect, tasks with similar words, terms, or entities (persons, documents, resources) will be clustered together based on semantic characteristics. For example, the content clusterer **123** is operable to create two clusters of events when it is noticed that a user performs certain tasks when working at a first location during a first time period and performs other tasks when working at a second location during a second time period to inform the heuristic engine **121** that there are two clusters of activity types regularly performed by the user. The content clusterer **123** enables the suggestion engine **122** to

provide suggested tasks that are appropriate for a given time and/or location at which those tasks are presented to the user.

**[0020]** For example, the user will be presented with task items related to work on days associated with the work week and business hours, but will be presented with task items related to domestic activities (e.g., clean bathroom, go shopping, groom dog) outside of business hours. In another example, the user will be presented with tasks related to work when located at the user's place of work (e.g., detected via Global Positioning System (GPS), Internet Protocol (IP) Location Services, network names in range of the user device **110**) and domestic tasks when located at another location (e.g., home, the grocery store, the dog groomer). In various aspects, the suggestion engine **122** will place various weights on clustering determinations that may change over a period of time, so that as time progresses, more or less weight will be given to the clustered content's location, time, or semantic data to allow for blended suggestions. For example, as the workday draws to a close, the user may be presented fewer work related tasks for the day as suggestions, and more domestic related tasks (e.g., "pick up milk on the way home from work"). In another example, when a location or a time period unknown to the content clusterer **123** is observed by the suggestion engine **122**, the suggestion engine **122** may rely on the other contextual data used to cluster tasks, such as, when a user is on vacation (in a location previously unknown to the suggestion engine **122**), the suggestion engine **122** may rely on time context and semantic context to provide suggestions, and ignore locational context.

**[0021]** A preview generator **124** is operable to generate previews for entities associated with a suggested task (or a selected task). For example, a portion of a document that is to be completed as part of a task is extracted by the preview generator **124** for presentation in a user interface as a preview. In another example, a portion of an audio recording of a phone call that is related to a task is generated as a preview. In a further example, a person who is related to a task (as a resource, an assignor, a teammate, or object of the task) has a preview generated with information from the relational graph service **130**, such as, for example, that person's contact information, an image of that person, biographical details of that person, etc.

**[0022]** User profiles **125** are stored by the task list service **120** so that as the behaviors of the users are observed by the heuristic engine **121**, the observations are stored to provide an increasingly more accurate view of the user's habits and use patterns for predicting future behaviors. In various aspects, the user or an administrator may also

manually set preferences in the user profiles **125** to define how tasks are to be presented to the user and aid the heuristic engine **121** in determining the user's preferences in addition to observing the user's actions to learn those preferences.

**[0023]** A context listener **126** is operable to receive (or request) contextual data and task items from the user device **110** and the services **130-160** for use by the task list service **120**. In various aspects, these data include appointments, events, meetings, and tasks set for the user and/or accepted by the user in addition to when and where these appointments, events, meetings, and tasks were set, accepted, worked on, and/or completed. In some aspects, the context listener **126** is operable to provide the state of the computing device (e.g., what applications were active, which application resulted in interacting with the task) to the task list service **120**. For example, metadata related to whether a user has looked at a given entity part of a task, how long the user has worked on a given task, how long it took between accepting the task and starting or completing the task, and what interactions were made by the user may be gathered for analysis and reporting.

**[0024]** A relational store **127** stores the relations observed for the creation of task items so that dynamic context can be provided to the user when the task is suggested to the user at a later date. For example, when the user manually or a system automatically creates a task item, the task is parsed to locate entities (e.g., persons involved, objects to be acted on) and recent actions (e.g., actions taken in the last m minutes) that may relate to the task item. For example, if the user receives a message containing the phrase "profit sharing plan" and creates a task that also include that phrase, a relationship between the task and the message will be formed and stored in the relational store **127**. In another example, when the user creates a task item to meet with another person, a relationship is formed between the task item, the meeting, and the person so that additional information about the meeting or the person can be recalled (e.g., from the relational graph service **130**) when the task item is presented to the user. In various aspects, the node identifiers from the relation graph service **130** for related entities are stored in the relational store **127**.

**[0025]** The relational graph service **130** hosts a graph database of a relational graph with nodes describing entities and a set of accompanying properties of those entities, such as, for example, the names, titles, ages, addresses, etc. Each property can be considered a key/value pair – a name of the property and its value. In other examples, entities represented as nodes that include documents, meetings, communication, etc., as well as edges representing relations among these entities, such as, for example, an edge

between a person node and a document node representing that person's authorship, modification, or viewing of the document. The relational graph service **130** executes graph queries that are submitted by various users to return nodes or edges that satisfy various conditions (e.g., users within the same division of a company, the last X documents accessed by a given user). In various aspects, the relational graph **130** is in communication with the other services **140-160** to match actions to documents and track edges between nodes representing entities from those other services **140-160**.

**[0026]** The email service **140** hosts the email communications for one or more users. In various aspects, the email service **140** is part of or includes a directory service for an organization. In other aspects, the email service **140** is integrated into or accessible by a productivity application of the productivity services **160**. For example, an email server storing email messages for an organization is accessible by email applications for members of that organizations and acts as an email service **140** accessible by the task list service **120**.

**[0027]** Emails provided from the email service **140** may be added as entities in the relational graph **130**, and/or the communications embodied by the emails are treated as edges between communicating parties. In various aspects, emails that are part of the tasks (e.g., "send an email to John Doe") that are monitored by the task list service **120**, and also provide context for other tasks, such as, for example, when a task is originated in an email (e.g., an email whose content includes "please review the meeting agenda" originates the task of "review meeting agenda").

**[0028]** The calendar service **150** hosts calendar and appointment information for one or more users. Various appointments, meetings, and events (collectively, events) are stored in the calendar service **150** that include one or more persons as participants/hosts. Events include one or more of: participants (required or optional), attendance information, times, locations, resources, attached documents, and event information (e.g., event title and description). In various aspects, the calendar service **150** is provided in a unified email/calendar application, such as, for example, THUNDERBIRD® (offered by the Mozilla Fnd of Mountain View, CA) or GMAIL® (offered by Alphabet Inc. of Mountain View, CA), which stores events for a user of that application. In other aspects, the calendar service **150** includes a social media platform, such as, for example, FACEBOOK® (offered by Facebook, Inc. of Menlo Park, CA) where various events are posted that users may attend.

**[0029]** Events provided from the calendar service **150** may be added as entities in the relational graph **130**, and/or the interactions embodied by the events are treated as edges between interacting parties. In various aspects, events are part of the tasks (e.g., "attend birthday party") that are monitored by the task list service **120**, and also provide context for other tasks, such as, for example, when a task is originated in an event (e.g., action items created during a meeting).

**[0030]** The productivity service **160** includes one or more productivity applications and document repositories that are accessible by one or more users. In various aspects, the productivity service **160** is hosted on the user device **110** and/or a remote server accessible by the user device **110**. For example, the productivity service **160** includes a locally executed authoring application (e.g., PAGES®, KEYNOTE®, or NUMBERS® offered by Apple, Inc. of Cupertino, CA) and remotely executed authoring applications (e.g., the GOOGLE DOCS™ suite offered by Alphabet, Inc. of Mountain View, CA) that are accessible via a thin client or web browser. In another example, the productivity service **160** include a library of documents stored on the user device **110** as well as libraries stored on networked computers or as part of a document management system and remote storage locations (e.g., GOOGLE DRIVE™ offered by Alphabet, Inc. of Mountain View, CA).

**[0031]** Documents provided from the productivity service **160** may be added as entities in the relational graph **130**. In various aspects, documents are part of the tasks (e.g., "edit the quarterly report") that are monitored by the task list service **120**, and provide context to report on how tasks have been handled to an initiating or collaborating party. For example, when a manager assigns the task of "edit the quarterly report" to an employee, the manager may receive an indication when the employee has completed the task, and the interactions that comprise that task. Similarly, when a manager assigns the task to a work group of several employees, when one employee assumes the task (e.g., begins work, accepts the task, completes the task), the other employees may be notified that the task has been assumed by their coworker.

**[0032]** In various aspects, the services **130-160** are operable to transmit interactions to the task list service **120** or to have interactions listened to/pulled from the services **130-160** to the task list service **120**. An API (Application Program Interface) or agent between the task list service **120** and services **130-160** facilitate communication between the services **130-160** and the task list service **120**, ensuring communications are received in a format interpretable by the receiving service. In one example, the SIRI® or GOOGLE NOW® personal digital assistants (offered by Apple, Inc. and Alphabet, Inc.,

respectively) may parse the sources **130-160** as agents to report relevant data to the task list service **120**. In another example, the sources **130-160** are configured to communicate to the task list service **120** as actions are taken in those services **130-160** in a format specified via an API.

**[0033]** **FIGURES 2A-10B** illustrate various example user interfaces for a task list application employing the tasks list service **120** to improve the efficiency in how tasks are presented to the user. As will be appreciated, **FIGURES 2A-10B** are provided as non-limiting examples and other arrangements and groupings of user interface elements with different content are included in the concepts of the present disclosure.

**[0034]** **FIGURES 2A** and **2B** are illustrations of example graphical user interfaces in which a Task List is implemented. The task list user interface illustrated in **FIGURE 2A** displays the scheduled tasks for "today". In one aspect, along with the task item, a link to the application relevant in completing the task item is provided. For example, the second item on the task list user interface displayed in **FIGURE 2A**, is "review notes for launch". A link to the notes application is provided along with the task item to allow the user to access the notes application from the task list application, instead of the user having to search for the application and then open it.

**[0035]** In one aspect, a link to the content item relevant to completing the task item is provided. For example, the first task item is "prepare screens for presentation". The task item is provided along with the content item "product_launchdeck" to allow the user to access the content item "product_launchdeck" in the presentation application without having to remember the content item and its location to complete the task item "prepare screens for presentation".

**[0036]** In one example, the tasks for "today" are listed in the order of time when they are due. In another example, they are listed in the order of priority. According to an example, the priority is identified by the system. In another example, the user is allowed to provide the priority details when creating the task item.

**[0037]** According to an aspect, the task list user interface illustrated in **FIGURE 2B** displays a suggested task list item. As illustrated, a suggested task list item "Book flight to San Francisco" is provided. According to an aspect, the task is suggested based on user context. User context may be developed by the system gradually by learning user patterns and user interaction data. The option to add the suggested task item to the task list shown in **FIGURE 2A** is provided. According to another aspect, the link provides a brief description of the task list item. For example, as illustrated in **FIGURE 2B**, a brief

description of the city of San Francisco is provided along with the suggested task list item "Book flight to San Francisco". In another example, others options such as for example, an option to provide "directions" to the city and an option to "book tickets" are provided.

**[0038]** As illustrated in **FIGURE 3**, the system displays suggested tasks for user to focus on. In the example illustrated in **FIGURE 3**, the first task list item "Check in for your flight" is provided as a focus item for the user. The system identifies a flight reservation in one or more applications accessed by the user, such as the mail application in which the flight confirmation may have arrived. Based on these identified data, the system provides the task list item "Check in for your flight".

**[0039]** According to another aspect, the system reviews the task list and suggests a task item that may not be due today, as a focus item. For example, if the system identifies a meeting scheduled for Friday, and the task item "prepare for meeting" is scheduled for Wednesday. The system may further identify that there are more task items scheduled for Wednesday than on Tuesday, and the system uses these data to provide the task item, "prepare for meeting" on Tuesday as focus task list item instead of on Wednesday.

**[0040]** As illustrated in **FIGURE 4A**, a calendar application showing one or more events (appointments, meetings, or events) is illustrated in relation to a calendar and order of occurrence for a day.

**[0041]** As illustrated in **FIGURE 4B**, the system provides a Plan of tasks for the user based on data received from the calendar application illustrated in **FIGURE 4A**. As illustrated, the task items that were not completed yesterday from the tasks application are provided to the user today. Further, an interface allowing the user to add a new task is provided. In one aspect, as illustrated in **FIGURE 4B**, the system provides the suggested tasks to the user based on data learned from one or more productivity applications (e.g., the calendar application shown in **FIGURE 4A**) by digital agents, for example, SIRI® (offered by Apple, Inc. of Cupertino CA) or ECHO® (offered by Amazon.com, Inc. of Seattle, WA), email applications, calendar applications, etc. For example, the suggestion task of "call grandma" is provided from the data learned from the calendar of a task item "Grandma Ruth's Birthday".

**[0042]** **FIGURE 5** is an illustration of an example graphical user interface in which Task List implementation is provided. In one aspect, the task list provides a list of task categories. For example as illustrated in **FIGURE 5**, the category list of tasks is provided on the left, showing several categories and subcategories (e.g., "To-Do", "Groceries", "School Play", Launch Event", "Presentation", "Reminders", and "Movies to

watch"). Upon selection of one or more categories, such as "presentation", the task items associated with the category – in this example, "presentation" – are provided in the middle of the illustrated graphical user interface. Further, additional details for task items such as due date, start date, reminders, descriptions, etc., that are associated with the task list, are provided along with the task list items, to the right of those items in the illustrated example.

**[0043]** **FIGURE 6** is an illustration of a graphical user interface in which a task application provides the task list items in relation to a task list category along with additional data about the selected task list item. For example, as illustrated in **FIGURE 6** the task items for the task list category "Movies to watch" are displayed for the user. The movies that may be recently watched and completed in the task list are marked completed and shown with a strike though, a check mark, or other indicia of their completion or rejection. Additional information about the selected task item (a movie in this example) is also provided. The additional information may include a brief description of the task item, an option of an application that can be used to interact with the task item, a link to access additional information about the task item in an online encyclopedia or relational graph, etc. In one example, the system provides the user with the due date data and allows the user to add the selected task item to a time period list (e.g., "today's tasks"). In one example, an option to create a new task item is also provided. Further, another option to delete the selected task item is also provided.

**[0044]** **FIGURE 7** is an illustration of a graphical user interface in which task list templates are provided to allow the users to get specifics information about a content item provided as a task item template, such as, for example, a job or a location. These task list templates can be used repeatedly for multiple users that are new to that job or location. As a result of task list application and the template's availability across all major platforms, it makes accessing actionable information at the right time and place easier for the user, improving their productivity. For example, the job "ABC-CITICO-450" illustrated in **FIGURE 7** is associated with a template task list, and the task list items included in the "ABC-CITICO-450" task list template help a user working on it to access relevant information in fewer steps and with lower risk of omitting a step.

**[0045]** **FIGURE 8** illustrates various source attributions for suggested tasks items. As shown in **FIGURE 8**, the task sources from which suggested tasks are drawn and the agent or suggestion engine **122** that is used to identify those suggested tasks is presented to the user. Various icons and source names are also presented in association with the

suggested tasks items to alert the user as to where, and potentially why, a given suggested task item is presented as a suggestion. For example, text from an email message may be extracted as promising a task item that is discovered by a context listener **126** and is therefore presented with an email icon along with the text that cause the context listener **126** to infer that a task item exists. In another example, a task in a user's task list for a later date is presented with a task list icon, and is suggested due to a surplus of inferred time on the user's schedule for the current date (e.g., today is open and the task item is due tomorrow). In a further example, a calendar icon is presented in association with an inferred task list item discovered from a calendar application.

**[0046]** **FIGURE 9** illustrates an example user interface for a digital assistant to create task items. As illustrated in **FIGURE 9**, a digital assistant has been invoked and provides various controls for defining a task item. In various aspects, user input may be received by the digital assistant from input devices such as keyboards, mice, and touch screens, but also from microphones when the digital assistant is operable to interpret speech for content input and commands. The digital assistant may be invoked by the user uttering a command sequence to summon the digital assistant, selecting a control in a user interface, or automatically in response to user actions in a productivity application, task list application, or operating system.

**[0047]** Input fields include, but are not limited to, title, description, persons involved, places involved, and times involved fields. The user is operable to set which task list the task item is added to, or the system may automatically add the task item to a task list according to a determination of common subject matter, time, or location according to a clusterer **123**. Additional controls are provided for the user to accept the creation of the task item (e.g., "remind"), reject the creation of the task item (e.g., "cancel"), and to locate additional data related to the task item (e.g., "search for …").

**[0048]** **FIGURES 10A** and **10B** illustrate example user interfaces for a time period view of a set of task items as the time period's task list is constructed. **FIGURE 10A** illustrates a starting position for a time period, a daily task list in the present example, which is blank when the time period list initiates. The blank view allows the user control over which candidate tasks for the day appear on the day's task list; suggested task items are shown but are not included on the list until selected by the user. In various aspects, the blank view is presented to the user at the start of the time period or at the conclusion of the prior time period. Although a user may pre-plan several tasks, in various aspects, the time period view is only accessible in the time period directly before or during the time period

associated with the time period list (e.g., that day or the day before for a daily list, at the start of a week or the end of the prior week for a weekly list, etc.).

**[0049]** Proceeding from **FIGURE 10A**, the interface illustrated in **FIGURE 10B** shows several suggested tasks for the time period as they are provided to the user. As shown in an upper portion in the interface, tasks from the prior time period (e.g., yesterday in a daily task list) are shown to the user so that incomplete tasks may be selected for inclusion in the current time period task list or the user is reminded of what was completed tasks were accomplished in the prior time period to create ongoing tasks. As shown in a middle portion of the interface, tasks that are overdue are shown to the user so that overdue tasks items may be selected for inclusion in the current time period task list. As shown in a lower portion of the interface, suggested task items from one or more services **130-160** as selected by a suggestion engine **120** are shown to the user to select from for inclusion in the time period list.

**[0050]** Various details about the suggested task items are shown to the user, including, without limitation: a title, a description, interested or relevant parties (e.g., assignor, assignee, sender, receiver, resource), due dates, start dates, portion already completed, sub-tasks, and related objects. Controls are provided in the interface in association with the suggested tasks to select one or more of the suggested task items to add an existing task list or new task list. Controls are also provided for the user to manually add task items to an existing or new task list. In other aspects, controls are provided to reject suggested task items, and the suggestion engine **122** is operable, in some aspects, to replace the rejected task items with other suggested task items. The heuristic engine **121** is operable to learn the user's behavior based on the user's interactions (e.g., selection, rejecting, ignoring) with the presented task items to improve the task items that the suggestion engine **122** provides.

**[0051]** **FIGURE 11** is a flow chart showing general stages involved in an example method **1100** for enhancing efficiency by clustering task items by their interaction contexts to provide more relevant results to users at different times and locations. The user's behavior is learned, based on how the user has interacted with the task list application, to provide tasks reminders or suggested tasks at a given time/location based on that learned behavior relative to the user's current context, time and location. For example, before going to work, the user may be at home, and because inputting work tasks into the task list application that may be based on the semantic context of the task items and/or historical data indicating that task items input at this time and location may be

completed at a different time and location (e.g., at the office), it may be determined to display those task items during work hours and/or at a work location. Similarly, although the user is at home and inputting tasks outside of work hours associated with the user, suggested tasks for work items are presented by the suggestion engine **122** based on the user pre-populating a work task list before heading into work.

**[0052]** Method **1100** begins at OPERATION **1110**, where task items are clustered based on similar times of interaction, locations of interaction, and content interacted with (e.g., semantically parsing emails or documents for key terms) the task items or related contextual objects. As will be appreciated, various clustering algorithms are used to determine any distinct groupings of task items, such as, for example, workday professional tasks, workday personal tasks, and weekend tasks. The system learns these clusters, but it does not need to know of their context (e.g., home versus work), just that certain tasks are interacted with in certain ways in certain locations and at certain times. The clusterer **123** clusters those task lists, task items, and keywords/concepts together into semantic concepts so that when the user is in the location/at the time when the task application is loaded, or uses related terminology, the heuristic engine **121** has learned over time based on user input that certain task items fall into certain clustered categories.

**[0053]** For example, a user with a task list labeled "songs I want to learn on guitar" is noted as interacting with the task list while at a first location (e.g., home) and during non-working hours will have that task list clustered towards being relevant while at the first location (e.g., home) and during non-working hours. Machine learning techniques, such as vector space search techniques, LDA (Latent Dirichlet Allocation) modeling, etc., are used to automatically create clusters on key words (e.g., task item names, document titles), their frequency of use, and multidimensional geographic proximity to determine that clusters and/or tasks are related. For example, when a user has input music related terms for task items, those terms will tend to overlap more frequently in text so that the clusterer **123** will determine that documents including terms, such as, for example, "notes," "songs," "intonation," etc., related to music and also relate to the task list "songs I want to learn on guitar," even though those terms might not all exist at the same time in each related document.

**[0054]** Additionally, as a user may group a plurality of task items into a task list, the clusterer **123** is further operable to determine that one task item belongs to a given cluster that the other task items on that task list should also belong to that given cluster. For example, for a "grocery shopping" task list, the user may input various foods to buy as

individual task items. If the clusterer **123** has sufficient semantic context to add a first task item (e.g., "buy milk") to a given cluster, but insufficient semantic context to add a second task item (e.g., "buy artichokes") to that cluster, the presence of the two tasks in the same task list enable the clusterer **123** to cluster them together, despite any insufficiency in semantic context. The clusterer **123** is enabled to learn from this grouping so that if the user were to create a new "grocery shopping list" without the first task item, but with the second task item, the second task item (and the remaining task items from the task list) will be assigned the semantic cluster for "grocery shopping".

**[0055]** At OPERATION **1120**, the time and location of accessing the task list application is determined. A user accesses the task list application not only by calling up a task list in a dedicated interface, but leaving a task list application active (e.g., as a background processes) to provide notifications, alerts, or alarms to provide contextually relevant tasks items in response to the location or time. In various aspects, the location of the user device **110** is determined based on GPS data from a GPS transponder of the user device **110** (if one is installed), IP location services for the IP address of the user device **110**, and/or the identity of the network to which the user device **110** is connected or that are available to connect to. The time of access is determined in various aspects based on a system clock included in the user device **110** or referencing an external time source.

**[0056]** Proceeding to OPERATION **1130**, relevant clusters of tasks for the current time of access and/or location of the user device **110** are identified. The time and a location associated with each cluster created in OPERATION **1110** is compared to the time and locations that are current for the user device **110** to match these times and locations to events in the task list or to present a contextually appropriate task list (or prevent a contextually inappropriate task list from being displayed). For example, a user who is checking a task list while at work may not want to have personal tasks displayed (e.g., due to including embarrassing personal details, distracting from work-related tasks), but will want those task items displayed when the user is not at work. In another aspect, as time progresses, it may be determined to provide tasks that belong in a cluster for the next time period.

**[0057]** For example, if the user is at a work location leading up to the end of a time associated with work activities, it may be determined to begin showing tasks associated with a non-work location and non-working times. A transition from one cluster to another may be sudden, or it may be gradual, in which a few items relevant to a second cluster are provided during a time/location of a first cluster, and are gradually replaced from the first

cluster as time progresses. For example, as the user's normal workday draws to the close, or extends into overtime, tasks items relevant to personal tasks will begin to replace workplace tasks, despite the user remaining in a work-location.

**[0058]** In another example, where a user is at a work location at a non-work related time, such as when going into on a weekend, it may be determined to show the user tasks associated with the work location, despite the time of access being a non-work related time. Depending on user preferences (e.g., stored in a user profile **125**), when a time context and a location context are matched to different clusters, tasks from both clusters may be identified for presentation to the user, tasks from clusters of a preferred context (locational or temporal) are identified for presentation to the user, or a context preference is used to weight which clusters have more (or fewer) tasks selected for presentation to the user. In various aspects, as candidate task items are presented to the user when an ambiguous context (i.e., the time context and the location context are matched to different clusters) is detected, the user is enabled to accept (or reject) the candidate task items, which will provide more (or fewer tasks) items from the indicated cluster.

**[0059]** When a location for a user device **110** is not recognized as corresponding to an existing cluster, the task list service **120** is operable to fall back to time as the only context to select a given cluster. The clusterer **123** tracks the new location and if there are enough data points in the new location, a new location cluster may be developed or an existing cluster may be associated with one or more (new) locations. For example, if the user leaves a workplace early and continues working at a coffee shop that has not be visited before, the cluster **123** is initially operable to use the time (e.g., "work hours") to provide work related tasks to the user. As the user's actions are monitored, the coffee shop's new location may be associated with a new task list, an existing working task list, or a subset of an existing working task list's task items to reflect the user's behavior at the coffee shop relative to task items. For example, if the user leaves a workplace and replies to several emails towards the end of the workday or writes documents (free from workplace distractions), the clusterer **123** is operable to note the tasks that the user performs at that environment and/or time.

**[0060]** Each cluster has a known context, such that the clusterer **123** interprets that a user is at a first location, a second location, etc., during a first time range, a second time range, etc., to identify different clusters based on when the users are interacting with the task list application and the services **130-160**. The clusterer **123** does not need to know the identity of these locations (e.g., workplace, home, gym, coffee shop, etc. are treated as

locations one, two, three, four, etc.) or time ranges (e.g., workhours, off-hours, weekend, weekday, etc. are treated as time ranges one, two, three, four, etc.), but a user may supply context for these clustering criteria that the clusterer **123** may use to identify various activities or clusters (e.g., office during work hours, office afterhours, home-headed to work, home-headed to gym, home-returned).

**[0061]** In another aspect, a new cluster may be developed when a new time period (or new actions in an already seen time period) is observed, in which the task list service **120** is operable to fall back to location as the only context to select a given cluster. For example, a user may begin waking up earlier to incorporate a new exercise regime before leaving for work. In another example, a user may carve out a portion of the day for performing a new set of tasks, such as, when the user decides for personal growth to start doing research on a certain topic every Tuesday and Thursday during a lunch hour normally associated with "work hours" while located at a location associated as a "workplace." The clusterer **123**, in coordination with the heuristic engine **121**, learns that there is a time slot where there is a cluster of tasks that are distinct from the rest of the tasks and will learn that Tuesday and Thursday during lunch is a special time for the user, even though the location has not changed.

**[0062]** Method **1100** proceeds to OPERATION **1140** where tasks from the relevant clusters, based on the available clusters having one or more contextual data for time of access or location of access in common with the current time of access or location of access are presented to the user device **110** for display and/or suggestion via the task list application. In various aspects, the task items from a relevant cluster come from one or more task lists (e.g., an "after work hours" cluster includes a "household chores" list and a "grocery shopping" list) and all or a portion of the available task items. A task list, or a title thereof, is presented in addition or instead of the task items thereon. For example, a task list for "grocery shopping" may be presented instead of or in addition to component task items of "buy milk" and "buy bread", and a title of a task list may be substituted for its components tasks when at least a user-configurable number of task items from that list are selected for presentation as candidate task items. Method **1100** may then conclude.

**[0063]** While implementations have been described in the general context of program modules that execute in conjunction with an application program that runs on an operating system on a computer, those skilled in the art will recognize that aspects may also be implemented in combination with other program modules. Generally, program

modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types.

**[0064]** The aspects and functionalities described herein may operate via a multitude of computing systems including, without limitation, desktop computer systems, wired and wireless computing systems, mobile computing systems (e.g., mobile telephones, netbooks, tablet or slate type computers, notebook computers, and laptop computers), hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, and mainframe computers.

**[0065]** In addition, according to an aspect, the aspects and functionalities described herein operate over distributed systems (e.g., cloud-based computing systems), where application functionality, memory, data storage and retrieval and various processing functions are operated remotely from each other over a distributed computing network, such as the Internet or an intranet. According to an aspect, user interfaces and information of various types are displayed via on-board computing device displays or via remote display units associated with one or more computing devices. For example, user interfaces and information of various types are displayed and interacted with on a wall surface onto which user interfaces and information of various types are projected. Interaction with the multitude of computing systems with which implementations are practiced include, keystroke entry, touch screen entry, voice or other audio entry, gesture entry where an associated computing device is equipped with detection (e.g., camera) functionality for capturing and interpreting user gestures for controlling the functionality of the computing device, and the like.

**[0066]** **FIGURES 12-14** and the associated descriptions provide a discussion of a variety of operating environments in which examples are practiced. However, the devices and systems illustrated and discussed with respect to **FIGURES 12-14** are for purposes of example and illustration and are not limiting of a vast number of computing device configurations that are utilized for practicing aspects, described herein.

**[0067]** **FIGURE 12** is a block diagram illustrating physical components (i.e., hardware) of a computing device **1200** with which examples of the present disclosure may be practiced. In a basic configuration, the computing device **1200** includes at least one processing unit **1202** and a system memory **1204**. According to an aspect, depending on the configuration and type of computing device, the system memory **1204** comprises, but is not limited to, volatile storage (e.g., random access memory), non-volatile storage (e.g., read-only memory), flash memory, or any combination of such memories. According to an

aspect, the system memory **1204** includes an operating system **1205** and one or more program modules **1206** suitable for running software applications **1250**. According to an aspect, the system memory **1204** includes the task list service **120**. The operating system **1205**, for example, is suitable for controlling the operation of the computing device **1200**. Furthermore, aspects are practiced in conjunction with a graphics library, other operating systems, or any other application program, and are not limited to any particular application or system. This basic configuration is illustrated in **FIGURE 12** by those components within a dashed line **1208**. According to an aspect, the computing device **1200** has additional features or functionality. For example, according to an aspect, the computing device **1200** includes additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated in **FIGURE 12** by a removable storage device **1209** and a non-removable storage device **1210**.

**[0068]** As stated above, according to an aspect, a number of program modules and data files are stored in the system memory **1204**. While executing on the processing unit **1202**, the program modules **1206** (e.g., task list service **120**) perform processes including, but not limited to, one or more of the stages of the method **1100** illustrated in **FIGURE 11**. According to an aspect, other program modules are used in accordance with examples and include applications such as electronic mail and contacts applications, word processing applications, spreadsheet applications, database applications, slide presentation applications, drawing or computer-aided application programs, etc.

**[0069]** According to an aspect, the computing device **1200** has one or more input device(s) **1212** such as a keyboard, a mouse, a pen, a sound input device, a touch input device, etc. The output device(s) **1214** such as a display, speakers, a printer, etc. are also included according to an aspect. The aforementioned devices are examples and others may be used. According to an aspect, the computing device **1200** includes one or more communication connections **1216** allowing communications with other computing devices **1218**. Examples of suitable communication connections **1216** include, but are not limited to, radio frequency (RF) transmitter, receiver, and/or transceiver circuitry; universal serial bus (USB), parallel, and/or serial ports.

**[0070]** The term computer readable media, as used herein, includes computer storage media. Computer storage media include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, or program modules.

The system memory **1204**, the removable storage device **1209**, and the non-removable storage device **1210** are all computer storage media examples (i.e., memory storage.) According to an aspect, computer storage media include RAM, ROM, electrically erasable programmable read-only memory (EEPROM), flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other article of manufacture which can be used to store information and which can be accessed by the computing device **1200**. According to an aspect, any such computer storage media is part of the computing device **1200**. Computer storage media do not include a carrier wave or other propagated data signal.

**[0071]** According to an aspect, communication media are embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and include any information delivery media. According to an aspect, the term "modulated data signal" describes a signal that has one or more characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency (RF), infrared, and other wireless media.

**[0072]** **FIGURES 13A** and **13B** illustrate a mobile computing device **1300**, for example, a mobile telephone, a smart phone, a tablet personal computer, a laptop computer, and the like, with which aspects may be practiced. With reference to **FIGURE 13A**, an example of a mobile computing device **1300** for implementing the aspects is illustrated. In a basic configuration, the mobile computing device **1300** is a handheld computer having both input elements and output elements. The mobile computing device **1300** typically includes a display **1305** and one or more input buttons **1310** that allow the user to enter information into the mobile computing device **1300**. According to an aspect, the display **1305** of the mobile computing device **1300** functions as an input device (e.g., a touch screen display). If included, an optional side input element **1315** allows further user input. According to an aspect, the side input element **1315** is a rotary switch, a button, or any other type of manual input element. In alternative examples, mobile computing device **1300** incorporates more or fewer input elements. For example, the display **1305** may not be a touch screen in some examples. In alternative examples, the mobile computing device **1300** is a portable phone system, such as a cellular phone. According to an aspect, the mobile computing device **1300** includes an optional keypad **1335**. According to an aspect,

the optional keypad **1335** is a physical keypad. According to another aspect, the optional keypad **1335** is a "soft" keypad generated on the touch screen display. In various aspects, the output elements include the display **1305** for showing a graphical user interface (GUI), a visual indicator **1320** (e.g., a light emitting diode), and/or an audio transducer **1325** (e.g., a speaker). In some examples, the mobile computing device **1300** incorporates a vibration transducer for providing the user with tactile feedback. In yet another example, the mobile computing device **1300** incorporates input and/or output ports, such as an audio input (e.g., a microphone jack), an audio output (e.g., a headphone jack), and a video output (e.g., a HDMI port) for sending signals to or receiving signals from an external device. In yet another example, the mobile computing device **1300** incorporates peripheral device port **1340**, such as an audio input (e.g., a microphone jack), an audio output (e.g., a headphone jack), and a video output (e.g., a HDMI port) for sending signals to or receiving signals from an external device.

**[0073]** **FIGURE 13B** is a block diagram illustrating the architecture of one example of a mobile computing device. That is, the mobile computing device **1300** incorporates a system (i.e., an architecture) **1302** to implement some examples. In one example, the system **1302** is implemented as a "smart phone" capable of running one or more applications (e.g., browser, e-mail, calendaring, contact managers, messaging clients, games, and media clients/players). In some examples, the system **1302** is integrated as a computing device, such as an integrated personal digital assistant (PDA) and wireless phone.

**[0074]** According to an aspect, one or more application programs **1350** are loaded into the memory **1362** and run on or in association with the operating system **1364**. Examples of the application programs include phone dialer programs, e-mail programs, personal information management (PIM) programs, word processing programs, spreadsheet programs, Internet browser programs, messaging programs, and so forth. According to an aspect, the task list service **120** is loaded into memory **1362**. The system **1302** also includes a non-volatile storage area **1368** within the memory **1362**. The non-volatile storage area **1368** is used to store persistent information that should not be lost if the system **1302** is powered down. The application programs **1350** may use and store information in the non-volatile storage area **1368**, such as e-mail or other messages used by an e-mail application, and the like. A synchronization application (not shown) also resides on the system **1302** and is programmed to interact with a corresponding synchronization application resident on a host computer to keep the information stored in

the non-volatile storage area **1368** synchronized with corresponding information stored at the host computer. As should be appreciated, other applications may be loaded into the memory **1362** and run on the mobile computing device **1300**.

**[0075]** According to an aspect, the system **1302** has a power supply **1370**, which is implemented as one or more batteries. According to an aspect, the power supply **1370** further includes an external power source, such as an AC adapter or a powered docking cradle that supplements or recharges the batteries.

**[0076]** According to an aspect, the system **1302** includes a radio **1372** that performs the function of transmitting and receiving radio frequency communications. The radio **1372** facilitates wireless connectivity between the system **1302** and the "outside world," via a communications carrier or service provider. Transmissions to and from the radio **1372** are conducted under control of the operating system **1364**. In other words, communications received by the radio **1372** may be disseminated to the application programs **1350** via the operating system **1364**, and vice versa.

**[0077]** According to an aspect, the visual indicator **1320** is used to provide visual notifications and/or an audio interface **1374** is used for producing audible notifications via the audio transducer **1325**. In the illustrated example, the visual indicator **1320** is a light emitting diode (LED) and the audio transducer **1325** is a speaker. These devices may be directly coupled to the power supply **1370** so that when activated, they remain on for a duration dictated by the notification mechanism even though the processor **1360** and other components might shut down for conserving battery power. The LED may be programmed to remain on indefinitely until the user takes action to indicate the powered-on status of the device. The audio interface **1374** is used to provide audible signals to and receive audible signals from the user. For example, in addition to being coupled to the audio transducer **1325**, the audio interface **1374** may also be coupled to a microphone to receive audible input, such as to facilitate a telephone conversation. According to an aspect, the system **1302** further includes a video interface **1376** that enables an operation of an on-board camera **1330** to record still images, video stream, and the like.

**[0078]** According to an aspect, a mobile computing device **1300** implementing the system **1302** has additional features or functionality. For example, the mobile computing device **1300** includes additional data storage devices (removable and/or non-removable) such as, magnetic disks, optical disks, or tape. Such additional storage is illustrated in **FIGURE 13B** by the non-volatile storage area **1368**.

**[0079]** According to an aspect, data/information generated or captured by the mobile computing device **1300** and stored via the system **1302** are stored locally on the mobile computing device **1300**, as described above. According to another aspect, the data are stored on any number of storage media that are accessible by the device via the radio **1372** or via a wired connection between the mobile computing device **1300** and a separate computing device associated with the mobile computing device **1300**, for example, a server computer in a distributed computing network, such as the Internet. As should be appreciated such data/information are accessible via the mobile computing device **1300** via the radio **1372** or via a distributed computing network. Similarly, according to an aspect, such data/information are readily transferred between computing devices for storage and use according to well-known data/information transfer and storage means, including electronic mail and collaborative data/information sharing systems.

**[0080]** **FIGURE 14** illustrates one example of the architecture of a system for improving the efficiency of managing task lists as described above. Content developed, interacted with, or edited in association with the task list service **120** is enabled to be stored in different communication channels or other storage types. For example, various documents may be stored using a directory service **1422**, a web portal **1424**, a mailbox service **1426**, an instant messaging store **1428**, or a social networking site **1430**. The task list service **120** is operative to use any of these types of systems or the like for improving efficiency of task list management, as described herein. According to an aspect, a server **1420** provides the task list service **120** to clients **1405a,b,c**. As one example, the server **1420** is a web server providing the task list service **120** over the web. The server **1420** provides the task list service **120** over the web to clients **1405** through a network **1440**. By way of example, the client computing device is implemented and embodied in a personal computer **1405a**, a tablet computing device **1405b** or a mobile computing device **1405c** (e.g., a smart phone), or other computing device. Any of these examples of the client computing device are operable to obtain content from the store **1416**.

**[0081]** Implementations, for example, are described above with reference to block diagrams and/or operational illustrations of methods, systems, and computer program products according to aspects. The functions/acts noted in the blocks may occur out of the order as shown in any flowchart. For example, two blocks shown in succession may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality/acts involved.

**[0082]** The description and illustration of one or more examples provided in this application are not intended to limit or restrict the scope as claimed in any way. The aspects, examples, and details provided in this application are considered sufficient to convey possession and enable others to make and use the best mode. Implementations should not be construed as being limited to any aspect, example, or detail provided in this application. Regardless of whether shown and described in combination or separately, the various features (both structural and methodological) are intended to be selectively included or omitted to produce an example with a particular set of features. Having been provided with the description and illustration of the present application, one skilled in the art may envision variations, modifications, and alternate examples falling within the spirit of the broader aspects of the general inventive concept embodied in this application that do not depart from the broader scope.

CLAIMS

1. A method for clustering task items by interaction contexts to provide more relevant results to users at different times and locations, comprising:

clustering task items for a user into clusters based on times and locations of the task items;

in response to a user device accessing a task list service, determining a time of access and a location of the user device at the time of access;

identifying relevant clusters from the clusters based on the time of access and the location of the user device; and

presenting task items from the relevant clusters on the user device.

2. The method of claim 1, wherein clustering the task items further comprises:

observing user actions relative to the task items to determine the times and the locations of the task items.

3. The method of claim 1, wherein clustering the task items further comprises:

observing key words of the task items; and

clustering the task items in association with the key words.

4. The method of claim 1, wherein clustering the task items further comprises:

observing task lists comprising pluralities of task items; and

in response to adding one task item from a given task list to a given cluster, adding remaining tasks items of the given task list to the given cluster.

5. The method of claim 1, wherein determining the location of the user device further is based on at least one of:

Global Positioning System data for the user device;

Internet Protocol location services related to an IP address of the user device; or

an identity of a network to which the user device is connected.

6. The method of claim 1, wherein identifying the relevant clusters further comprises:

receiving user input rejecting a given task item presented on the user device from a given relevant cluster; and

in response to the user input rejecting the given task item, presenting a new task item from a different relevant cluster.

7. The method of claim 1, wherein presenting task items from the relevant clusters further comprises:

determining whether a user-configurable number of the task items from the relevant clusters belong to a given task list; and

in response to determining that a user-configurable number of the task items from the relevant clusters belong to the given task list, substituting a name of the given task list for the task items for presentation on the user device.

8. A system for clustering task items by interaction contexts to provide more relevant results to users at different times and locations, comprising:

a processor; and

a computer readable memory storage device including instructions that when executed by the processor enable the system to:

cluster task items for a user into clusters based on times and locations of the task items;

in response to a user device accessing a task list service, determine a time of access and a location of the user device at the time of access;

identify relevant clusters from the clusters based on the time of access and the location of the user device; and

present task items from the relevant clusters on the user device.

9. The system of claim 8, wherein to cluster the task items the system is further operable to:

observe user actions relative to the task items to determine the times and the locations of the task items

10. The system of claim 8, wherein to cluster the task items the system is further operable to:

observe key words of the task items; and

cluster the task items in association with the key words.

11. The system of claim 8, wherein to cluster the task items the system is further operable to:

observe task lists comprising pluralities of task items; and

in response to adding one task item from a given task list to a given cluster, adding remaining tasks items of the given task list to the given cluster.

12. The system of claim 8, wherein to determine the location of the user device the system is uses at least one of:

Global Positioning System data for the user device;

Internet Protocol location services related to an IP address of the user device; or

an identity of a network to which the user device is connected.

13. The system of claim 8, wherein to identify the relevant clusters the system is further operable to:

receive user input rejecting a given task item presented on the user device from a given relevant cluster; and

in response to the user input rejecting the given task item, reject the given cluster as a relevant cluster.

14. The system of claim 8, wherein to present task items from the relevant clusters the system is further operable to:

determine whether a user-configurable number of the task items from the relevant clusters belong to a given task list; and

in response to determining that a user-configurable number of the task items from the relevant clusters belong to the given task list, substituting a name of the given task list for the task items for presentation on the user device.

15. A computer readable storage device including processor executable instructions for clustering task items by interaction contexts to provide more relevant results to users at different times and locations, comprising:

clustering task items for a user into clusters based on times and locations of the task items;

in response to a user device accessing a task list service, determining a time of access and a location of the user device at the time of access;

identifying relevant clusters from the clusters based on the time of access and the location of the user device; and

presenting task items from the relevant clusters on the user device.

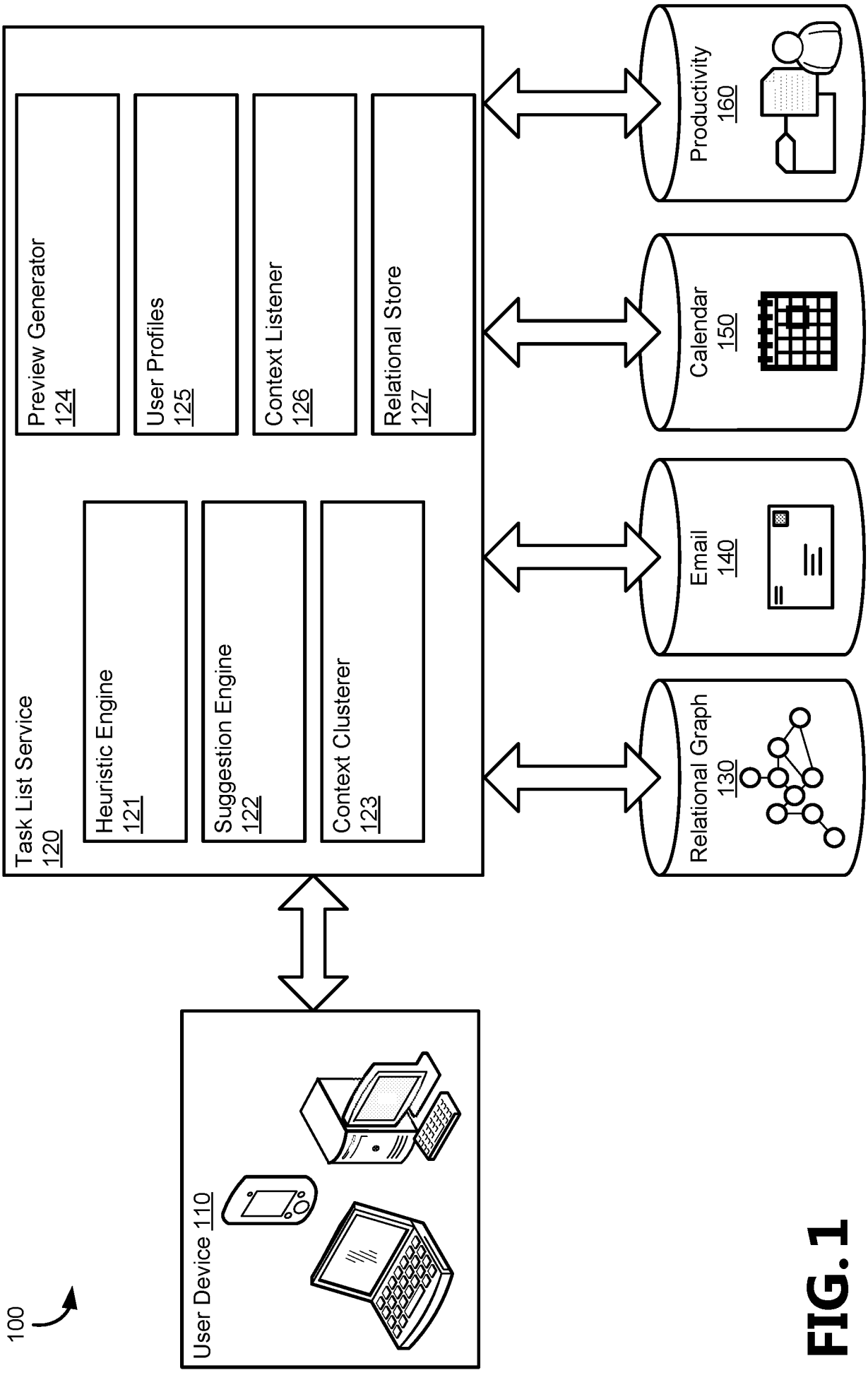

100
User Device 110
Task List Service 120
Heuristic Engine 121
Suggestion Engine 122
Context Clusterer 123
Preview Generator 124
User Profiles 125
Context Listener 126
Relational Store 127
Relational Graph 130
Email 140
Calendar 150
Productivity 160

FIG. 1

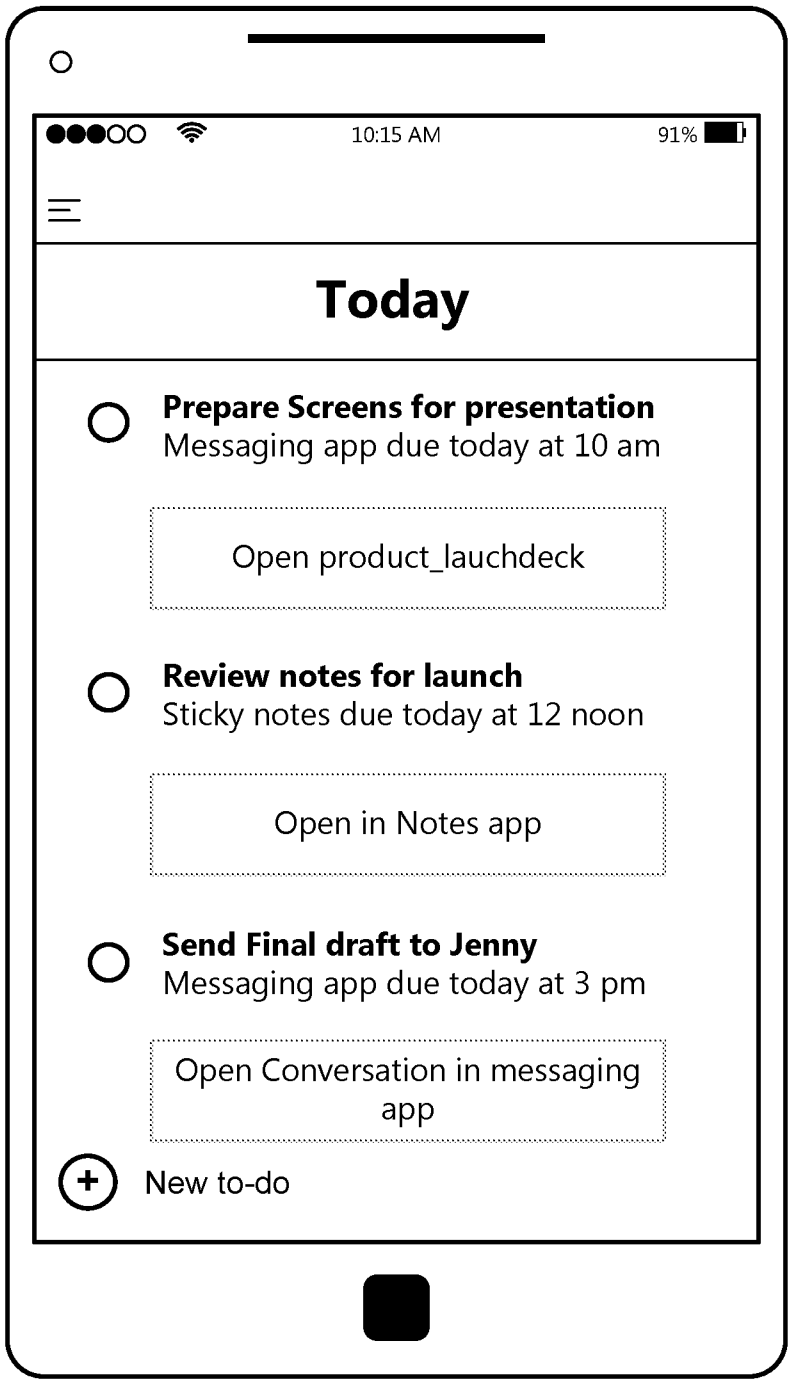

10:15 AM
91%
Today
Prepare Screens for presentation
Messaging app due today at 10 am
Open product_lauchdeck
Review notes for launch
Sticky notes due today at 12 noon
Open in Notes app
Send Final draft to Jenny
Messaging app due today at 3 pm
Open Conversation in messaging app
New to-do

FIG. 2A

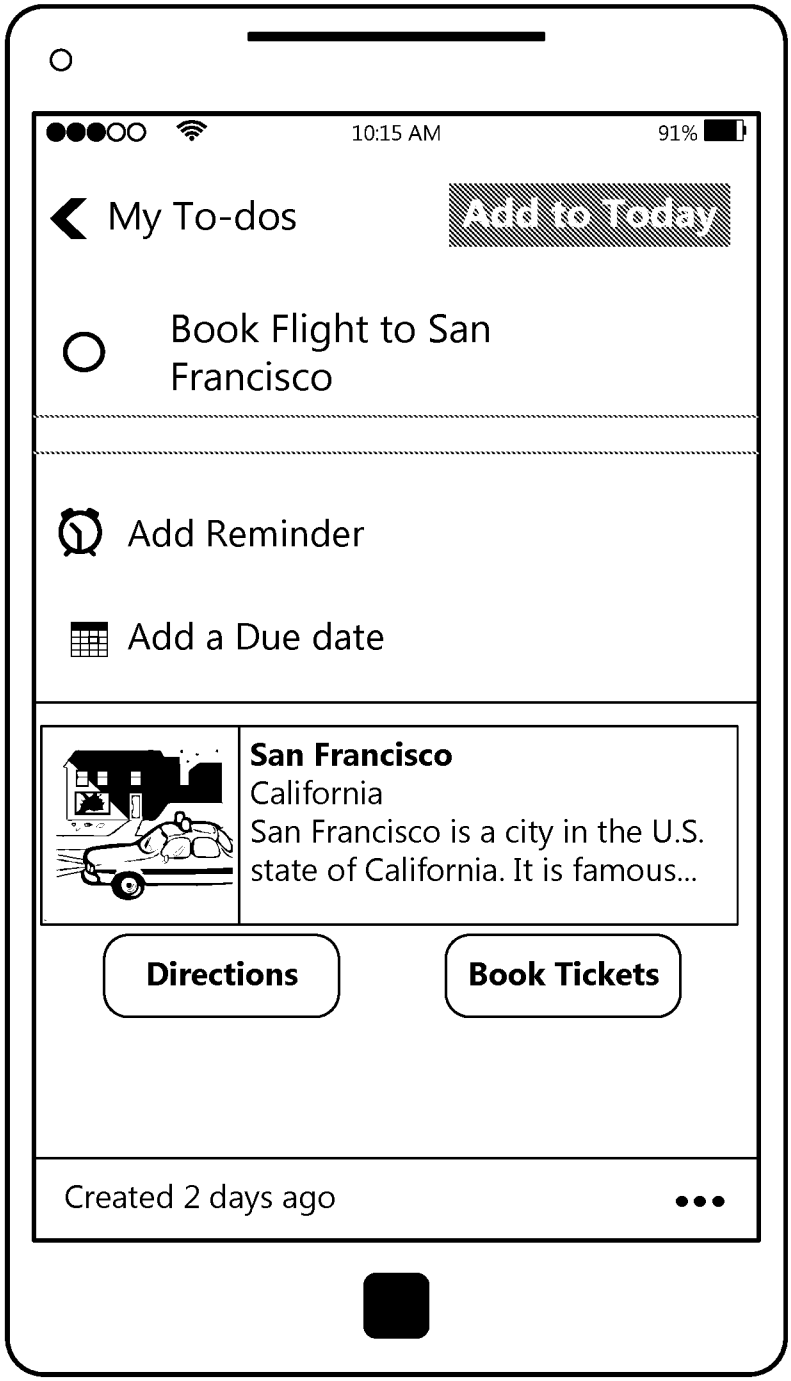

10:15 AM
91%
My To-dos
Add to Today
Book Flight to San Francisco
Add Reminder
Add a Due date
San Francisco
California
San Francisco is a city in the U.S. state of California. It is famous...
Directions
Book Tickets
Created 2 days ago

FIG. 2B

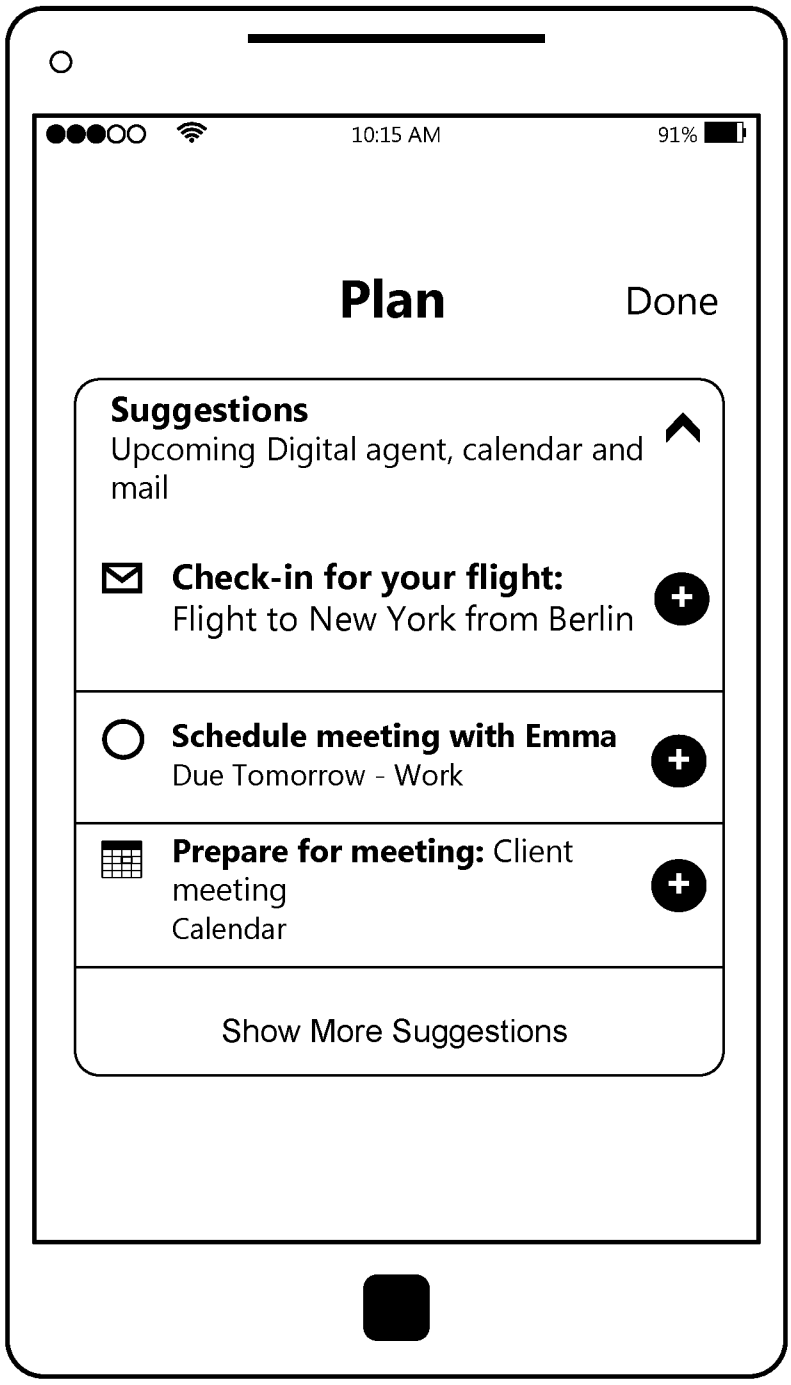

10:15 AM
91%
Plan
Done
Suggestions
Upcoming Digital agent, calendar and mail
Check-in for your flight:
Flight to New York from Berlin
Schedule meeting with Emma
Due Tomorrow - Work
Prepare for meeting: Client meeting
Calendar
Show More Suggestions

FIG. 3

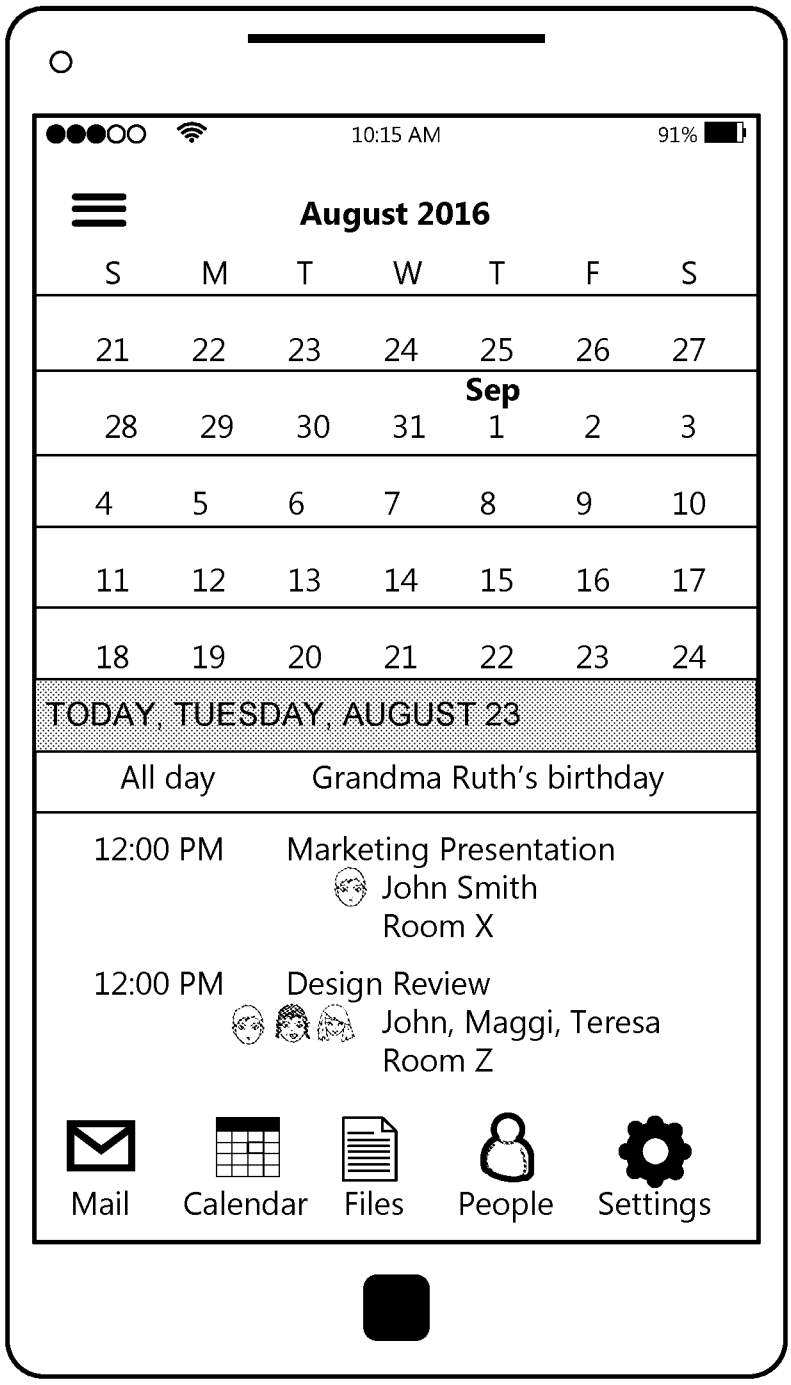

10:15 AM
91%
August 2016
S M T W T F S
21 22 23 24 25 26 27
Sep
28 29 30 31 1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
TODAY, TUESDAY, AUGUST 23
All day Grandma Ruth's birthday
12:00 PM Marketing Presentation
John Smith
Room X
12:00 PM Design Review
John, Maggi, Teresa
Room Z
Mail
Calendar
Files
People
Settings

FIG. 4A

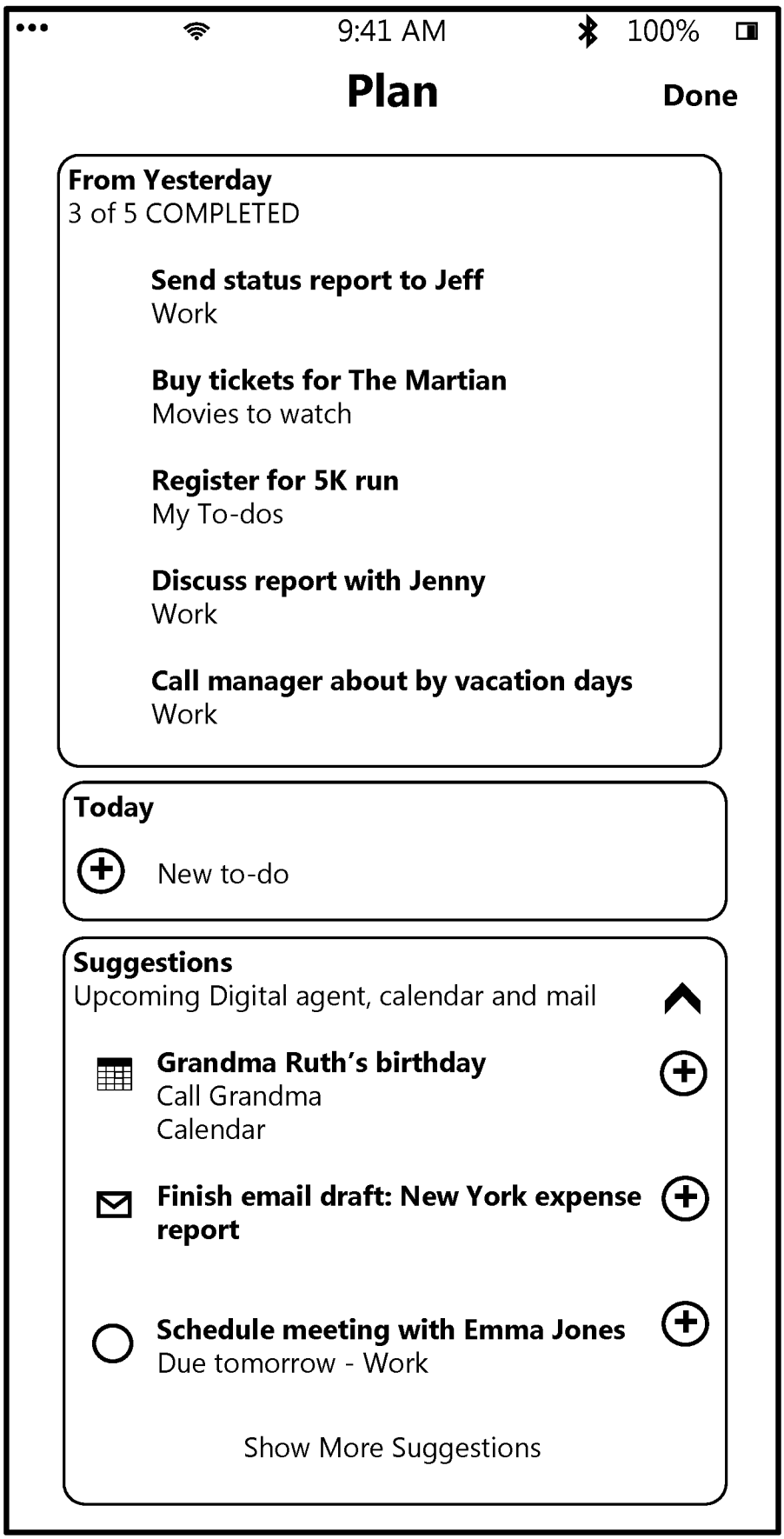

9:41 AM
100%
Plan
Done
From Yesterday
3 of 5 COMPLETED
Send status report to Jeff
Work
Buy tickets for The Martian
Movies to watch
Register for 5K run
My To-dos
Discuss report with Jenny
Work
Call manager about by vacation days
Work
Today
New to-do
Suggestions
Upcoming Digital agent, calendar and mail
Grandma Ruth's birthday
Call Grandma
Calendar
Finish email draft: New York expense report
Schedule meeting with Emma Jones
Due tomorrow - Work
Show More Suggestions

FIG. 4B

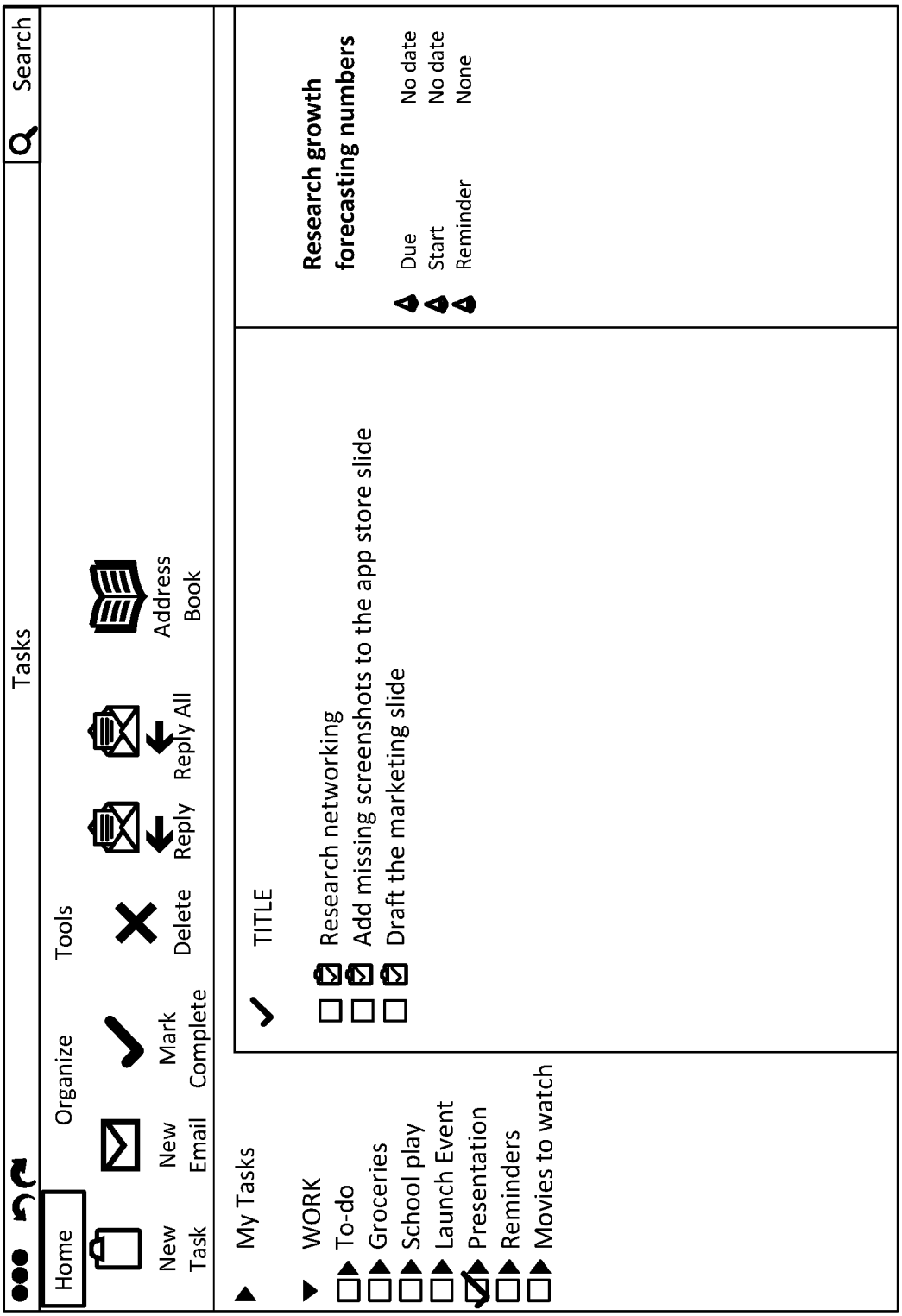

Tasks
Search
Home
Organize
Tools
New Task
New Email
Mark Complete
Delete
Reply
Reply All
Address Book
My Tasks
WORK
To-do
Groceries
School play
Launch Event
Presentation
Reminders
Movies to watch
TITLE
Research networking
Add missing screenshots to the app store slide
Draft the marketing slide
Research growth forecasting numbers
Due No date
Start No date
Reminder None

FIG. 5

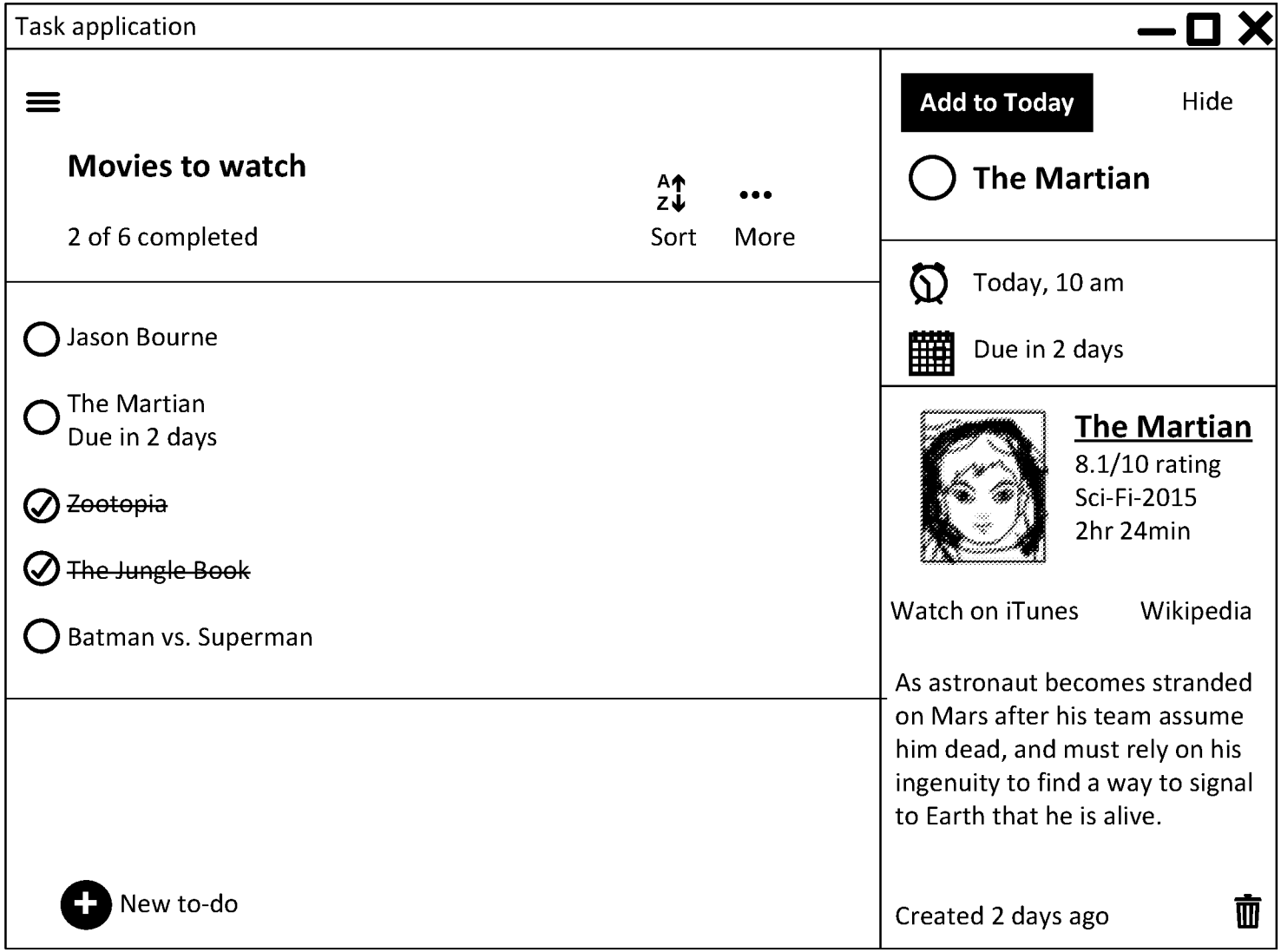

Task application
Movies to watch
2 of 6 completed
Sort
More
Jason Bourne
The Martian
Due in 2 days
Zootopia
The Jungle Book
Batman vs. Superman
New to-do
Add to Today
Hide
The Martian
Today, 10 am
Due in 2 days
The Martian
8.1/10 rating
Sci-Fi-2015
2hr 24min
Watch on iTunes
Wikipedia
As astronaut becomes stranded on Mars after his team assume him dead, and must rely on his ingenuity to find a way to signal to Earth that he is alive.
Created 2 days ago

FIG. 6

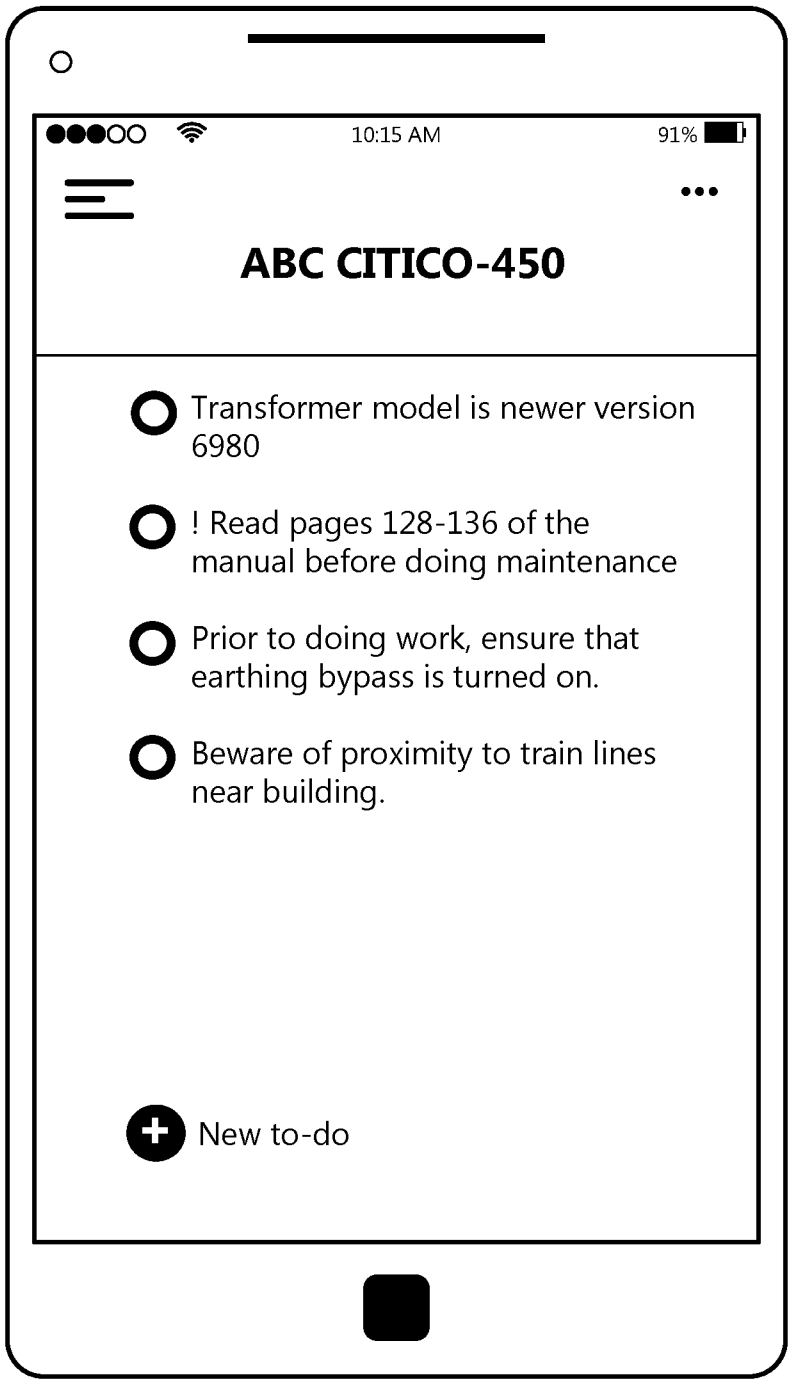

10:15 AM
91%
ABC CITICO-450
Transformer model is newer version 6980
! Read pages 128-136 of the manual before doing maintenance
Prior to doing work, ensure that earthing bypass is turned on.
Beware of proximity to train lines near building.
New to-do

FIG. 7

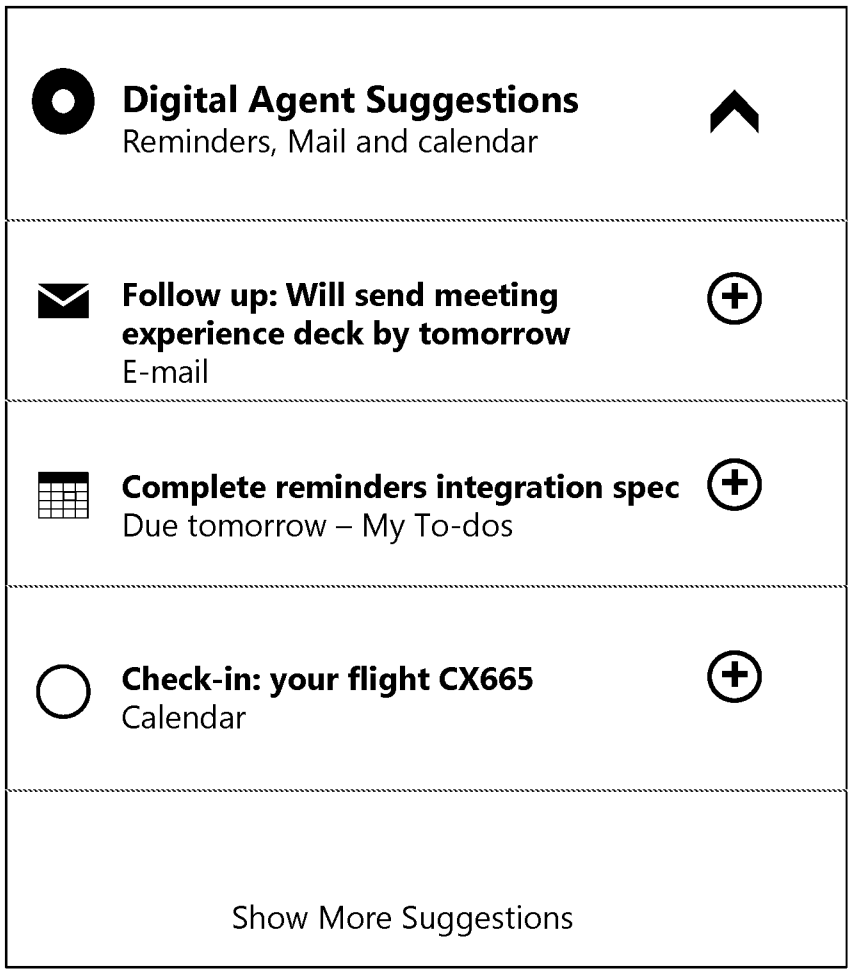

Digital Agent Suggestions
Reminders, Mail and calendar
Follow up: Will send meeting experience deck by tomorrow
E-mail
Complete reminders integration spec
Due tomorrow – My To-dos
Check-in: your flight CX665
Calendar
Show More Suggestions

FIG. 8


When should I remind you?
Reminder
Talk about growth projections
Launch Event
Person,
Place
or
Time
Remind
Cancel
Search for “Talk about growth Projections”
remind me to feed the cat

FIG. 9

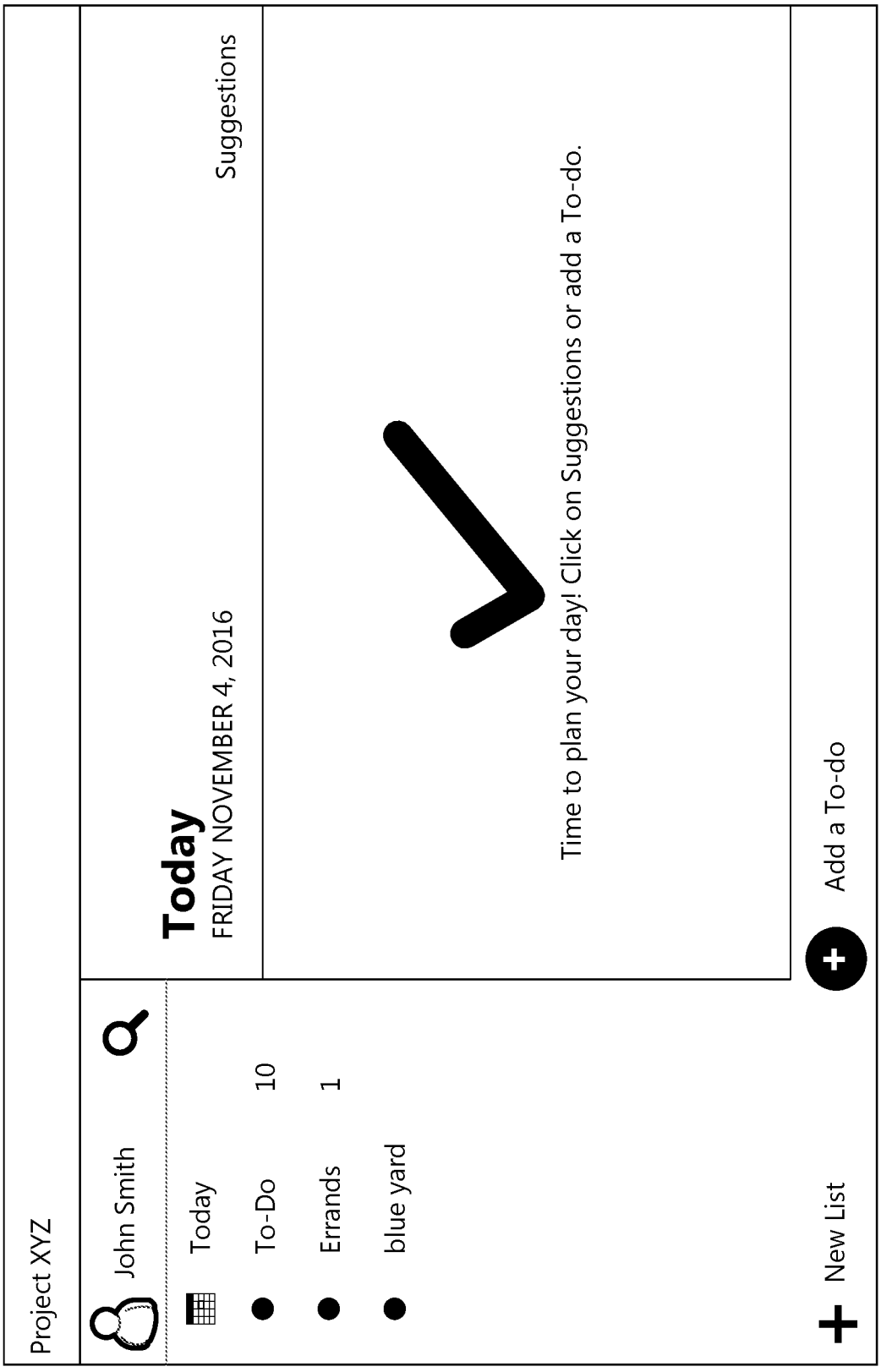

Project XYZ
John Smith
Today
To-Do 10
Errands 1
blue yard
New List
Today
FRIDAY NOVEMBER 4, 2016
Suggestions
Time to plan your day! Click on Suggestions or add a To-do.
Add a To-do

FIG. 10A

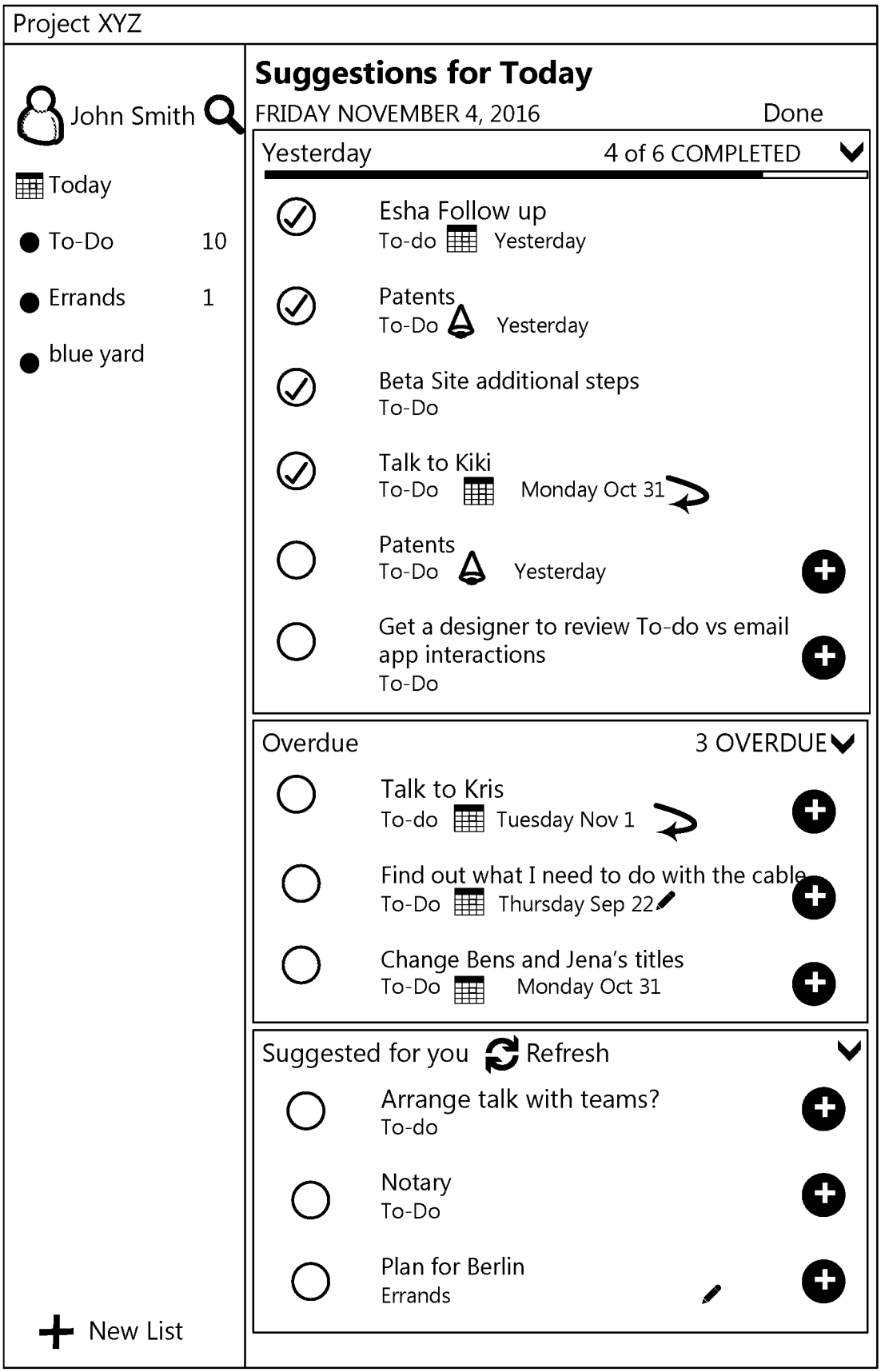

Project XYZ
John Smith
Today
To-Do 10
Errands 1
blue yard
New List
Suggestions for Today
FRIDAY NOVEMBER 4, 2016
Done
Yesterday
4 of 6 COMPLETED
Esha Follow up
To-do Yesterday
Patents
To-Do Yesterday
Beta Site additional steps
To-Do
Talk to Kiki
To-Do Monday Oct 31
Patents
To-Do Yesterday
Get a designer to review To-do vs email app interactions
To-Do
Overdue
3 OVERDUE
Talk to Kris
To-do Tuesday Nov 1
Find out what I need to do with the cable
To-Do Thursday Sep 22
Change Bens and Jena's titles
To-Do Monday Oct 31
Suggested for you
Refresh
Arrange talk with teams?
To-do
Notary
To-Do
Plan for Berlin
Errands

FIG. 10B

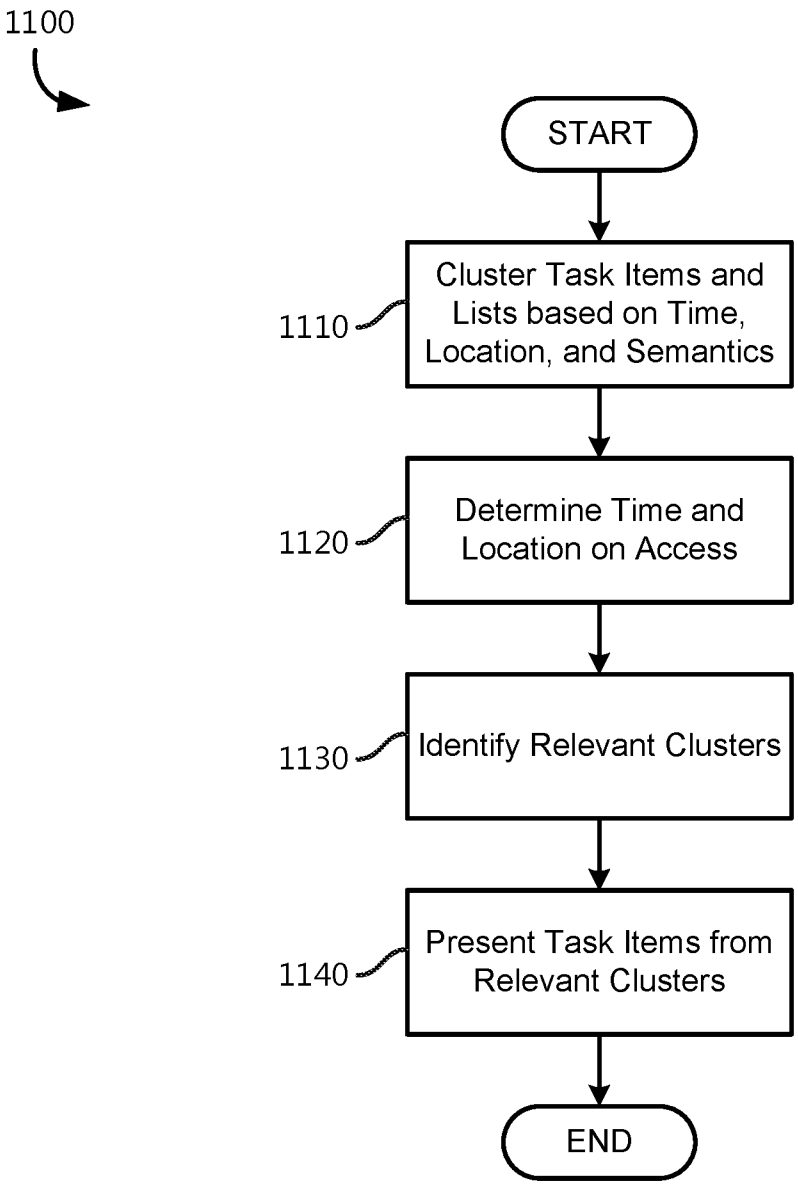

1100
START
Cluster Task Items and Lists based on Time, Location, and Semantics
1110
Determine Time and Location on Access
1120
Identify Relevant Clusters
1130
Present Task Items from Relevant Clusters
1140
END

FIG. 11

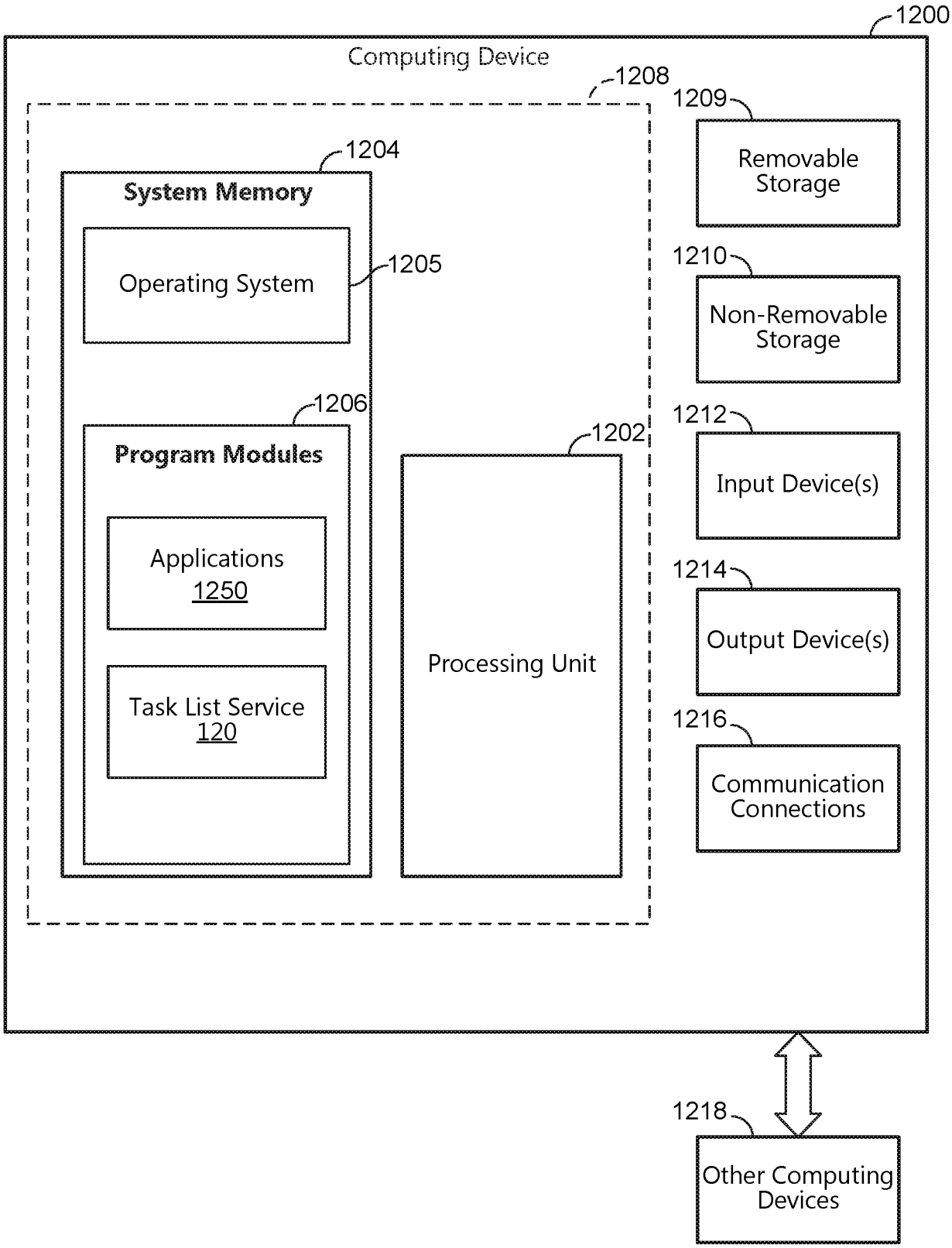
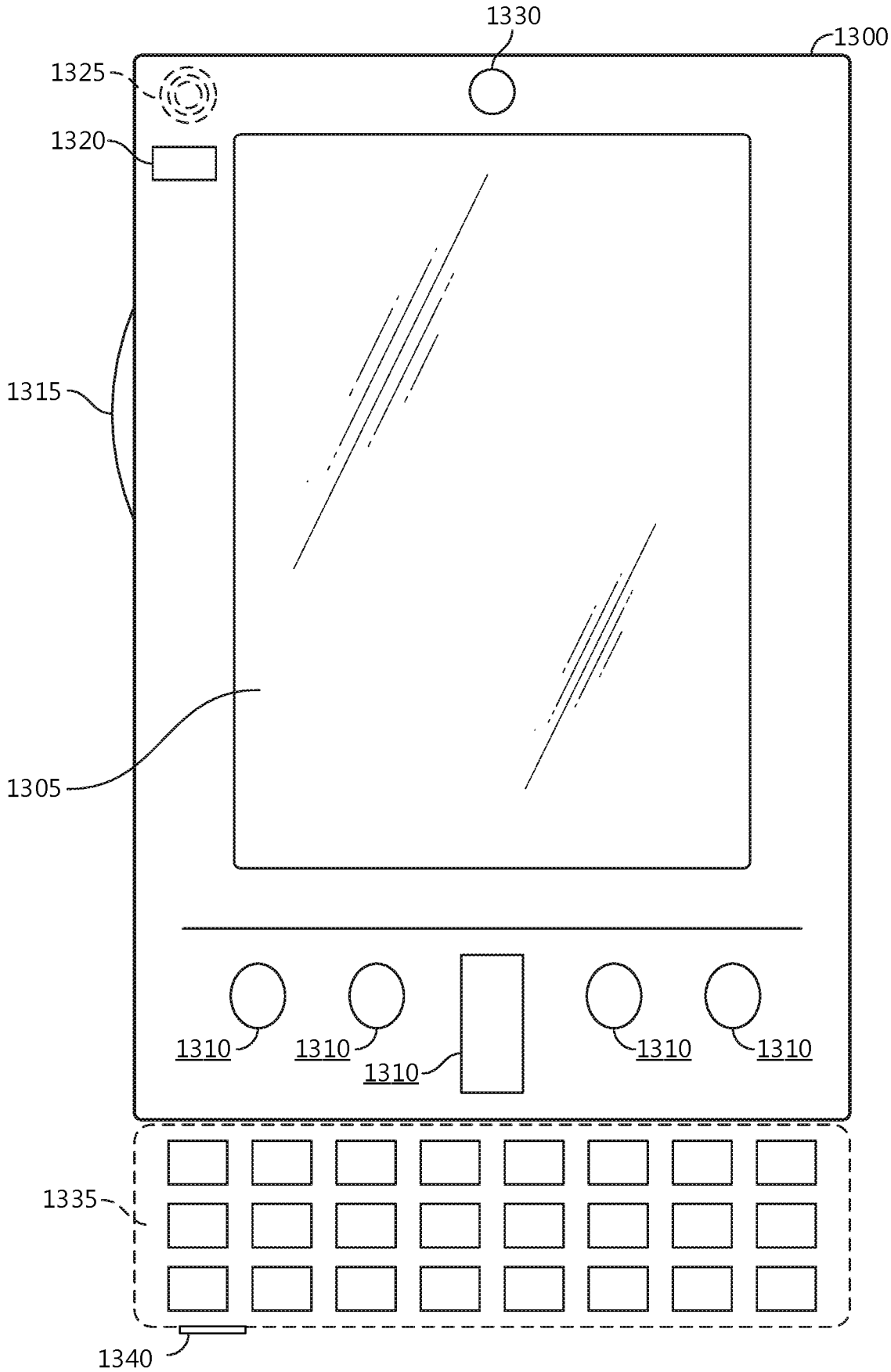

1200
Computing Device
1208
1209
Removable Storage
1204
System Memory
Operating System
1205
1210
Non-Removable Storage
1206
Program Modules
1202
1212
Input Device(s)
Applications 1250
Processing Unit
1214
Output Device(s)
Task List Service 120
1216
Communication Connections
1218
Other Computing Devices

FIG. 12




MOBILE COMPUTING DEVICE

FIG. 13A

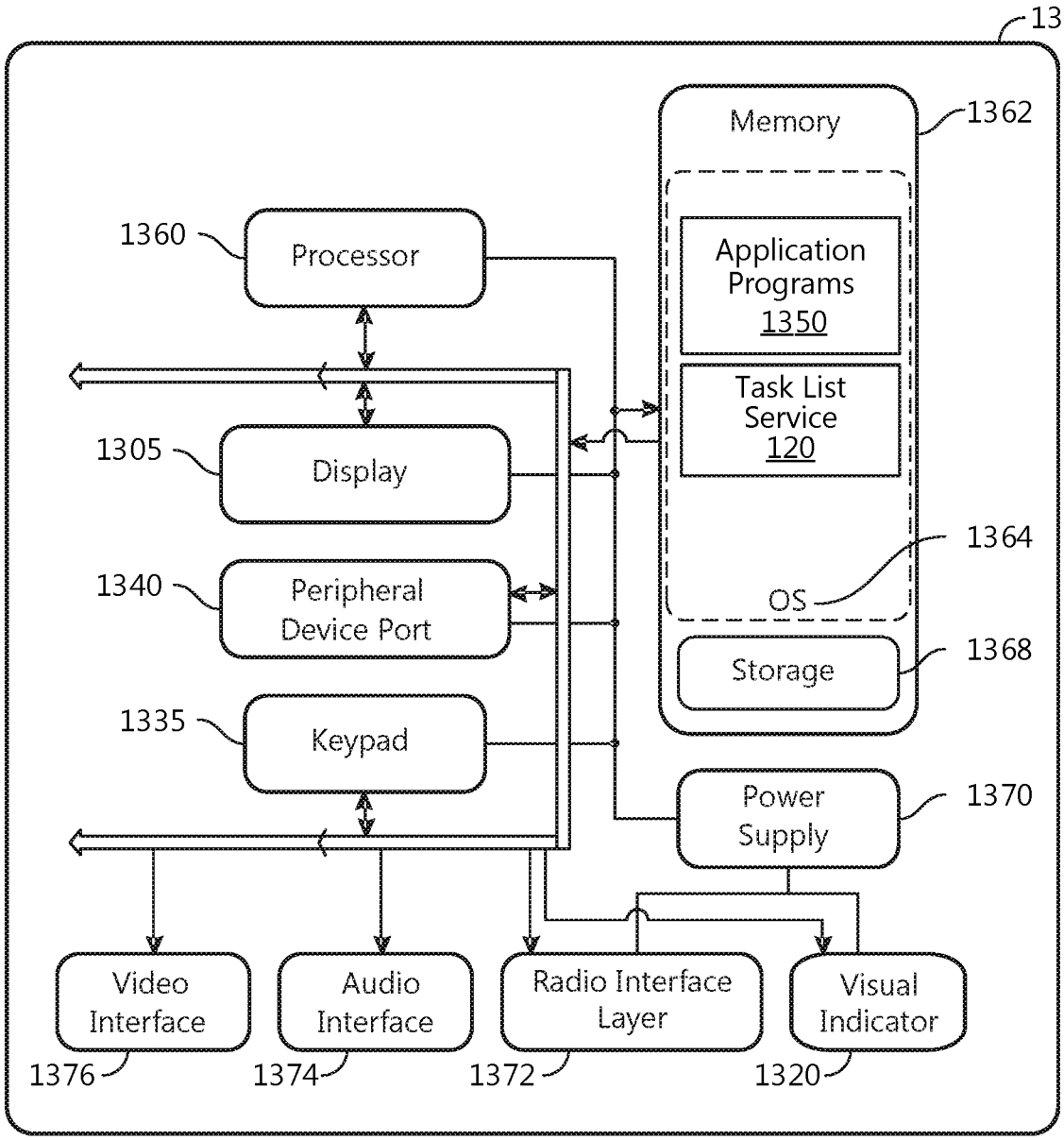

1302
Memory
1362
1360
Processor
Application Programs 1350
Task List Service 120
1305
Display
1364
1340
Peripheral Device Port
OS
1368
Storage
1335
Keypad
Power Supply
1370
Video Interface
Audio Interface
Radio Interface Layer
Visual Indicator
1376
1374
1372
1320

FIG. 13B

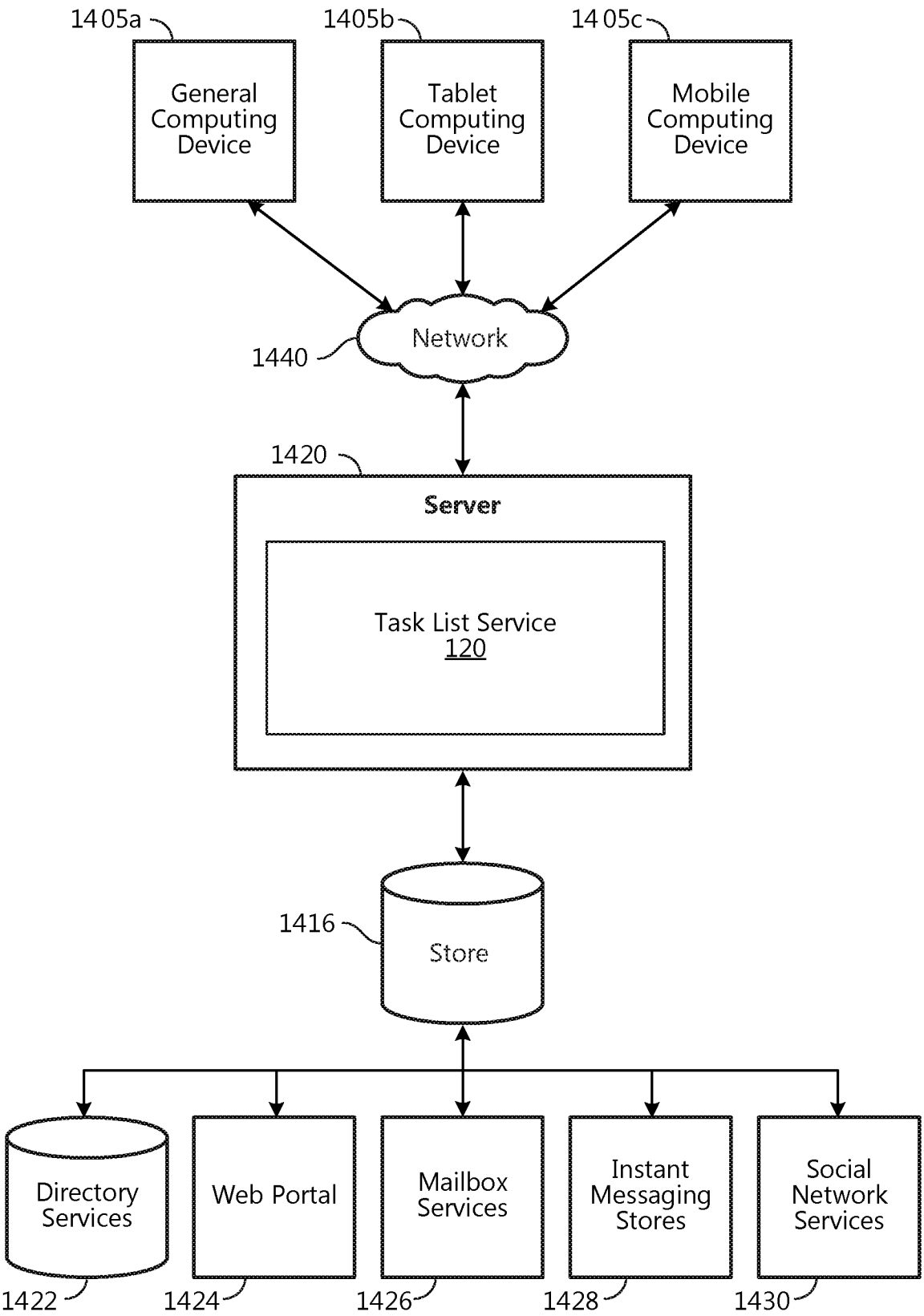

1405a
General Computing Device
1405b
Tablet Computing Device
1405c
Mobile Computing Device
Network
1440
1420
Server
Task List Service
120
1416
Store
Directory Services
Web Portal
Mailbox Services
Instant Messaging Stores
Social Network Services
1422
1424
1426
1428
1430

FIG. 14

# INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/059636

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06Q10/10 G06F17/30
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06Q G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | US 2006/282298 A1 (AZVINE BEHNAM [GB] ET AL) 14 December 2006 (2006-12-14) paragraphs [0097] - [0122]; figure 2a | 1-15 |
| X | US 2016/086116 A1 (RAO SUPRIYA [IN] ET AL) 24 March 2016 (2016-03-24) paragraphs [0018] - [0024], [0030] - [0035], [0049] - [0051]; figures | 1-15 |
| X | US 2014/173602 A1 (KIKIN-GIL EREZ [US] ET AL) 19 June 2014 (2014-06-19) paragraphs [0020] - [0064]; figures | 1-15 |

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 19 December 2017 | 08/01/2018 |

| Name and mailing address of the ISA/ | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | Blackley, William |

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2017/059636

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 2006282298 A1 | 14-12-2006 | CA 2534843 A1 | 17-03-2005 |
| | | EP 1665143 A1 | 07-06-2006 |
| | | US 2006282298 A1 | 14-12-2006 |
| | | WO 2005024683 A1 | 17-03-2005 |
| US 2016086116 A1 | 24-03-2016 | NONE | |
| US 2014173602 A1 | 19-06-2014 | AR 094045 A1 | 08-07-2015 |
| | | CN 105027077 A | 04-11-2015 |
| | | EP 2932375 A2 | 21-10-2015 |
| | | JP 2016502204 A | 21-01-2016 |
| | | KR 20150097504 A | 26-08-2015 |
| | | TW 201435759 A | 16-09-2014 |
| | | US 2014173602 A1 | 19-06-2014 |
| | | WO 2014093637 A2 | 19-06-2014 |