



- (51) **International Patent Classification:**
G06Q 10/06 (2012.01) G06F 17/30 (2006.01)
- (21) **International Application Number:**
PCT/US2017/059633
- (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
15/450,714 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, Benjamin 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, 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).

(54) **Title:** EFFICIENCY ENHANCEMENTS IN TASK MANAGEMENT APPLICATIONS

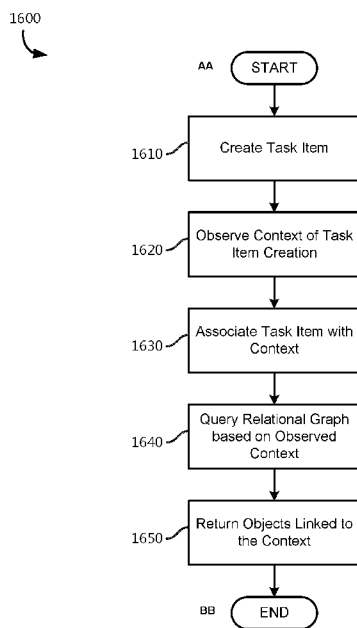


FIG. 16

(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.



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
5 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
10 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
15 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
20 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, dynamic context is provided for tasks to provide the user
25 with greater recall for details related to the task. The context in which the task item was
created is provided in a traceable format for the user to interpret the origins of a task item.
For example, a user with a task item for “call John Doe” may be provided with the entities
that resulted in the need to “call John Doe,” such as an email from John requesting a call,
and details related to John, such as John’s phone number, an image of John, etc. The
30 dynamic context connects tasks to the reasons why they were created and resources for
how they may be completed, and includes references for locations, times, persons,
documents, and other entities relevant to the task items for the user’s review.

[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.

5 [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.

10 [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

15 [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;

20 **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;

25 **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;

30 **FIGURE 6** illustrates an example graphical user interface in which Task List item creation from within a productivity application is shown;

FIGURE 7 illustrates a graphical user interface in which a Task List is implemented;

FIGURE 8 illustrates a graphical user interface in which a Task List is implemented within the productivity application in which it can be completed;

FIGURE 9 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 10 illustrates a graphical user interface in which an interface to
5 create a task list item is provided within a productivity application, such as a messaging application;

FIGURE 11 illustrates an additional option for creating a task list item within the example messaging application;

FIGURE 12 illustrates a graphical user interface in which the creation of a
10 task list item is shown within a communication application, such as, for example, a messaging application;

FIGURE 13 illustrates a graphical user interface in which a task list application template is provided;

FIGURES 14A and **14B** illustrate a graphical user interface in which a task
15 assigned by a first user can be provided to a second user in the application relevant to completing that task item;

FIGURE 15 illustrates an example user interface for a digital assistant to create task items;

FIGURE 16 is a flow chart showing general stages involved in an example
20 method for enhancing efficiency by providing dynamic context for tasks to provide the user with greater ability to recall the details and circumstances related to the tasks;

FIGURE 17 is a block diagram illustrating example physical components of a computing device;

FIGURES 18A and **18B** are block diagrams of a mobile computing device;
25 and

FIGURE 19 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
30 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.

5 [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
10 greater control and convenience in manipulating task items.

[0011] In one aspect, dynamic context is provided for tasks to provide the user with greater recall for details related to the task. The context in which the task item was created is provided in a traceable format for the user to interpret the origins of a task item. For example, a user with a task item for “call John Doe” may be provided with the entities
15 that resulted in the need to “call John Doe,” such as an email from John requesting a call, and details related to John, such as John’s phone number, an image of John, etc. The dynamic context connects tasks to the reasons why they were created and resources for how they may be completed, and includes references for locations, times, persons, documents, and other entities relevant to the task items for the user’s review.

20 [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.

25 [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
30 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 17, 18A, 18B, and 19**. 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, time, and semantics 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

5 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.

10 **[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 tasks 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

15 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

20 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

25 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

30 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
5 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
10 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-15** 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-15** are provided as non-
15 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
20 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
25 having to remember the content item and its location to complete the task item “prepare screens for presentation”. Content items include various file formats for authoring and/or
30 viewing content, such as, for example: word processing files, email files, calendaring files, spreadsheet files, database files, note taking files, presentation files, image files, audio files, video files, etc.

[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.

5 [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
10 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.

15 [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
20 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
25 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.

30 [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 an example graphical user interface in which Task List item creation from within a productivity application is shown. As is illustrated in **FIGURE 6**, a task creation option is provided in an email productivity application. In one example, the system identifies the “volunteer registration” as an entity possibly related to a task and identifies the candidate time as “tonight” and provides the task creation control. Upon selection of the task creation control, the “volunteer registration” task is created as a task item for “tonight”. In another aspect, the user selects the task creation option and identifies the objects, persons, and times relevant to that task. In various aspects, dialogs or prompts are provided to the user to select the values from the productivity application or manually input those values to build the task.

[0044] **FIGURE 7** is an illustration of a graphical user interface in which a Task List is implemented. In one aspect, upon selection of the Task creation option in **FIGURE 6**, the task item is created in relation to the content item from which it was created and is displayed in a calendar. For example, the system identifies the “volunteer registration” task in the email illustrated in **FIGURE 6** and that the task needs to be completed tonight – relative to the day of creation or a date in the content item. The system creates the task and displays the task in the calendar application as illustrated in **FIGURE 7** for

completion tonight. In one example, the user is provided with an option whether to accept the task or to make any desired changes to the task such as for example, change the date and or time, reassigned the task, etc.

[0045] **FIGURE 8** is an illustration of a graphical user interface in which a Task List is implemented within the productivity application in which it can be completed. For example, as illustrated in **FIGURE 8**, a presentation application is in use and the tasks from the tasks list that are identified by the system to be completed in the presentation application are displayed for the user therein. This allows the user to open the task application and view the tasks to be completed without switching to the task list application. In various aspects, the user is also linked to the document or section of the document, via a hyperlink in the user interface that is related to the task item, to improve user navigation to task-relation documents or portions thereof.

[0046] In another aspect, the user is allowed to add a task item, as illustrated in **FIGURE 8**, from the productivity application without having to open or give focus to the task list application for interacting with the task list items. For example, the user is allowed to add a task item to the task list from the presentation application without ever leaving the presentation application and without having to access the task list application to create a task list item.

[0047] **FIGURE 9** 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 9** 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.

[0048] **FIGURE 10** is an illustration of a graphical user interface in which an interface to create a task list item is provided within a productivity application, such as for

example, a messaging application. As illustrated in **FIGURE 10**, an option to create a task list item is provided within a messaging application for a user to select. In one example, once the system identifies a task item is to be created, the productivity application is parsed for relevant objects for inclusion in the task item. For example, a series of messages
5 in a conversation document of a messaging application are parsed to identify objects of “spy movie” and a task time “this weekend”, to prepopulate a task item. The user is provided an interface to accept or substitute different values for the prepopulated with an option to add the task list item in the task list application.

[0049] In another example, once the system identifies a task list item, for example,
10 “spy movie” and a task time “this weekend”, the system automatically adds the task list item to the tasks list application. Further, in one example, the user visits the task list application and views the task list categories of movies to watch to find the “spy movie” task list item and does not remember the context in which the movie was added to the task list. The system may provide a context along with a link to the particular messaging
15 conversation to the user, with the “spy movie” task list item, in the task list application.

[0050] **FIGURE 11** illustrates an additional option to create a task list item within the messaging application. For example, a user may select an option to add a task item from the productivity application via a control or menu, as illustrated in **FIGURE 10**, or a control may be provided contextually by the task list service **120** within the productivity
20 application in response to the user’s actions. For example, as illustrated in **FIGURE 11**, when a user is entering text into a field, a control to create a task based on the contents of the field may be provided so that the user does not need to access a menu or control as part of a ribbon, but instead is provided a contextual control proximate to where the user is interaction with the productivity application.

[0051] **FIGURE 12** is an illustration of a graphical user interface in which the creation of a task list item is shown within a communication application, such as, for example, a messaging application. The system provides an agent which can be added as an entity to the user’s contact list. For example, as illustrated in **FIGURE 12**, the user is allowed to create a task list item by sending a message to the task list agent or an agent
30 within the messaging application. Further, the agent sends a message to the user via the messaging application confirming that the task list item has been created. In another aspect, along with the confirmation message, the agent also provides additional information related to the task list item. For example, as illustrated in **FIGURE 12**, when the user requests the agent to create a task to “make a reservation for dinner at the famous

restaurant”, the system provides a confirmation message, via the messaging application, notifying the user that the task list item has been created along with additional relevant information, such as, for example, a brief review of the restaurant, the timing of the reservation, a rating from a popular review site, a link to access the website of the restaurant, and a link to access directions to the restaurant, etc.

[0052] **FIGURE 13** is an illustration of a graphical user interface in which a task list application template is provided. Task list templates allow for the task items of a given list to be presented in repeated circumstances. For example, one or more task list templates may be provided to easily share information between the employees of an organization. In another example, a template task list may be added to a new employee’s task list by the employee’s manager, wherein that template task list is provided to each new employee for onboarding purposes, and the manager may receive notifications of the status of the completion for that employee. As task list items are completed by the employee the tasks will be marked as complete for that employee, but another new employee, who was provided the same template list, will need to independently fill out the task items; despite being based on the same template, each user receives a unique task list.

[0053] In various aspects, the task list templates are set up manually. In other examples, task list templates are created automatically. For example, a grocery shopping task list template is created for a user based on the user’s previous task lists for grocery shopping so that frequently purchased items automatically appear as items on the task list. The user may manually add to the template task list or to the resultant individual task list (e.g., always buy bread (add to template), buy bread this time (add to individual task list)). In various aspects, task lists for repeated tasks include items with varying frequencies (e.g., every time the list is created, every other time the list is created, after n days since the last list was created) so that, for example, a template for grocery shopping includes a task item for “buy toothpaste” that appears as part of the template task list, but only appears on the grocery shopping task list for an individual reminder to go grocery shopping once per month, despite the user going grocery shopping (based on the template task list) four times per month.

[0054] **FIGURES 14A** and **14B** are illustrations of a graphical user interface in which a task assigned by a first user can be provided to a second user in the application relevant in completing that task item. For example, as illustrated in **FIGURE 14A**, the task list items related to the presentation application that will need to be completed within the presentation application are provided as soon as the second user accesses the

presentation application. Along with the task list item, additional information about the first user who assigned the task list item, due date, percentage completed is provided. In another example, as illustrated in **FIGURE 14B**, the system allows the user to add an assigned task list item to the user's own task list for later completion by selecting "Add",
5 to perform the task at that moment by selecting "do it now", or to delegate the task to another user by selecting "delegate". In one example, the system provides a notification to the first user (the task assignor) of the second user's selection to update the status or assignee of the task list item.

[0055] **FIGURE 15** illustrates an example user interface for a digital assistant to create task items. As illustrated in **FIGURE 15**, 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
10 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.

[0056] 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 ...").
20

[0057] **FIGURE 16** is a flow chart showing general stages involved in an example method **1600** for enhancing efficiency by providing dynamic context for tasks to provide the user with greater ability to recall the details and circumstances related to the task, as well as, supplemental information to augment the understanding of the task.
25

[0058] For example, a user with a task in a task list for "follow-up on the documents" will understand what documents are to be followed-up on and for whom and when the task item is added to the task list, but may forget those details as time progresses. To alleviate the loss of context for a task item, the task list service **120** automatically relates the context with the task so that as tasks are created, actions and entities that lead to the creation of the task are captured for later review. For example, if a user received an
30

email from “John” asking that the user send him a document, then the user (or the task list service **120** on the user’s behalf) may have created the task item from within the email or while the email recently held focus in an email client or operating system. The task list service **120** is thereby enabled to continuously track that context so that when the user
5 look at tasks in the task list application, the originating context will be shown as well as supplemental information. Continuing the above example, links to access the email that the task is related to; an excerpt, preview, or link to the documents to follow-up on; and the biographical information of the sender, receivers, or other parties mentioned in the email (or related to those parties in the relational graph service **130**) will be shown to give
10 the user more details on how the task item is to be completed.

[0059] In another aspect, content items that may add additional context to an existing task item are identified and used to supplement the context of the task item. Continuing the above example, when a user receives a status request from another party about a task item (e.g., an email, phone call, instant messenger message), such as “how
15 close are you to completing that document for John?,” the task list service **120** is operable to identify the task item in question and associate the status request with the task item. By associating a new content item with a task item, the user can access the task item from the example status request or later review the task item and see in its context that a status request was made as to its progress.

[0060] Machine learning techniques are employed to determine what information related to context objects linked to task items to provide to the user. The heuristic engine **121** is operable to learn and use semantically connected relationships to source and create relationships between tasks and objects in other systems and services **130-160** throughout the operating environment. For example, the heuristic engine **121** is operable to learn over
25 time how to filter the available contextual information to provide the contextual information that is desired by the user so that certain terms, such as, for example, docket numbers, project names, working groups, etc., are learned and used to link objects to provide dynamic context to the user’s task items.

[0061] Method **1600** begins at OPERATION **1610** in response to a task item being
30 created in the task list system **120** and proceeds to OPERATION **1620**, where the context of the task item creation is gathered. In various aspects, which applications and documents are open or have focus prior to task item creation are observed. In another aspect, events occurring when the task item is created are observed, such as, for example, whether the

user was in a meeting when the task was created and the task item is therefore (likely) an action item for that meeting and related to the other persons from the meeting.

[0062] Proceeding to OPERATION 1630, the task item is associated in a relational graph with the context observed in OPERATION 1620. In various aspects, the node identifiers from the relation graph service 130 for the context entities (e.g., documents/communications open at the time of task item creation, persons co-scheduled for an ongoing event at the time of task item creation) are stored in the relational store 127 in association with the node identifiers for the task items and are related to one another via edges in the relational graph.

[0063] At OPERATION 1640 the relational graph service 130 is queried for information related to the contextual entities related to the tasks item. The relational graph is spanned, beginning from either the node representing the given user or from the node representing the task item assigned to the given user, along the edges representing context relationships to the nodes representing events, documents, or persons observed as providing context. In various aspects, the query includes requests for the contextual entities stored in the relational store 127 as well as entities located within *n* nodes of a contextual entity from the relational store 127. For example, when two persons who are members of a group are noted as related to a task item, the other members of that group (who may not have been contextually related to the task item – they may not have been co-scheduled for the meeting that originated the task item) may be discovered in the relational graph and presented as related to the task item based on their inclusion in a shared group with the original two persons.

[0064] Proceeding to OPERATION 1650, information on the objects linked to the context are returned to the user. In various aspects, the information is returned as hyperlinks to the context objects or as previews for the context objects. For example, a user may be provided a hyperlink to a meeting, email, or document that is determined to have initiated a task item. In another example, a contact card for a person related to the task item is provided (showing an image or avatar for that person, biographical details, organization chart details, contact information, etc.). Method 1600 may then conclude.

[0065] 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.

[0066] The aspects and functionalities described herein may operate via a multitude of computing systems including, without limitation, desktop computer systems, 5 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.

[0067] In addition, according to an aspect, the aspects and functionalities described 10 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 15 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, 20 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.

[0068] **FIGURES 17-19** and the associated descriptions provide a discussion of a variety of operating environments in which examples are practiced. However, the devices 25 and systems illustrated and discussed with respect to **FIGURES 17-19** 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.

[0069] **FIGURE 17** is a block diagram illustrating physical components (i.e., hardware) of a computing device **1700** with which examples of the present disclosure may 30 be practiced. In a basic configuration, the computing device **1700** includes at least one processing unit **1702** and a system memory **1704**. According to an aspect, depending on the configuration and type of computing device, the system memory **1704** 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 **1704** includes an operating system **1705** and one or more program modules **1706** suitable for running software applications **1750**. According to an aspect, the system memory **1704** includes the task list service **120**. The operating system **1705**, for example, is suitable for controlling the operation of the computing device **1700**.
5 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 17** by those components within a dashed line **1708**. According to an aspect, the computing device **1700** has additional features or functionality. For example, according to an aspect, the computing
10 device **1700** 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 17** by a removable storage device **1709** and a non-removable storage device **1710**.

[0070] As stated above, according to an aspect, a number of program modules
15 **1706** and data files are stored in the system memory **1704**. While executing on the processing unit **1702**, the program modules **1706** (e.g., task list service **120**) perform processes including, but not limited to, one or more of the stages of the method **1600** illustrated in **FIGURE 16**, respectively. According to an aspect, other program modules are used in accordance with examples and include applications such as electronic mail and
20 contacts applications, word processing applications, spreadsheet applications, database applications, slide presentation applications, drawing or computer-aided application programs, etc.

[0071] According to an aspect, the computing device **1700** has one or more input
25 device(s) **1712** such as a keyboard, a mouse, a pen, a sound input device, a touch input device, etc. The output device(s) **1714** 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 **1700** includes one or more communication connections **1716** allowing communications with other computing devices
30 **1718**. Examples of suitable communication connections **1716** include, but are not limited to, radio frequency (RF) transmitter, receiver, and/or transceiver circuitry; universal serial bus (USB), parallel, and/or serial ports.

[0072] 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 **1704**, the removable storage device **1709**, and the non-removable storage device **1710** 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 **1700**. According to an aspect, any such computer storage media is part of the computing device **1700**. Computer storage media do not include a carrier wave or other propagated data signal.

[0073] 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.

[0074] **FIGURES 18A** and **18B** illustrate a mobile computing device **1800**, 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 18A**, an example of a mobile computing device **1800** for implementing the aspects is illustrated. In a basic configuration, the mobile computing device **1800** is a handheld computer having both input elements and output elements. The mobile computing device **1800** typically includes a display **1805** and one or more input buttons **1810** that allow the user to enter information into the mobile computing device **1800**. According to an aspect, the display **1805** of the mobile computing device **1800** functions as an input device (e.g., a touch screen display). If included, an optional side input element **1815** allows further user input. According to an aspect, the side input element **1815** is a rotary switch, a button, or any other type of manual input element. In alternative examples, mobile computing device **1800** incorporates more or fewer input elements. For example, the display **1805** may not be a touch screen in some examples. In alternative examples, the mobile computing device **1800** is a portable phone system, such as a cellular phone. According to an aspect, the

mobile computing device **1800** includes an optional keypad **1835**. According to an aspect, the optional keypad **1835** is a physical keypad. According to another aspect, the optional keypad **1835** is a “soft” keypad generated on the touch screen display. In various aspects, the output elements include the display **1805** for showing a graphical user interface (GUI), a visual indicator **1820** (e.g., a light emitting diode), and/or an audio transducer **1825** (e.g., a speaker). In some examples, the mobile computing device **1800** incorporates a vibration transducer for providing the user with tactile feedback. In yet another example, the mobile computing device **1800** 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 **1800** incorporates peripheral device port **1840**, 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.

[0075] **FIGURE 18B** is a block diagram illustrating the architecture of one example of a mobile computing device. That is, the mobile computing device **1800** incorporates a system (i.e., an architecture) **1802** to implement some examples. In one example, the system **1802** 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 **1802** is integrated as a computing device, such as an integrated personal digital assistant (PDA) and wireless phone.

[0076] According to an aspect, one or more application programs **1850** are loaded into the memory **1862** and run on or in association with the operating system **1864**. 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 **1862**. The system **1802** also includes a non-volatile storage area **1868** within the memory **1862**. The non-volatile storage area **1868** is used to store persistent information that should not be lost if the system **1802** is powered down. The application programs **1850** may use and store information in the non-volatile storage area **1868**, 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 **1802** 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 **1868** synchronized with corresponding information stored at the host computer. As should be appreciated, other applications may be loaded into the memory **1862** and run on the mobile computing device **1800**.

5 [0077] According to an aspect, the system **1802** has a power supply **1870**, which is implemented as one or more batteries. According to an aspect, the power supply **1870** further includes an external power source, such as an AC adapter or a powered docking cradle that supplements or recharges the batteries.

10 [0078] According to an aspect, the system **1802** includes a radio **1872** that performs the function of transmitting and receiving radio frequency communications. The radio **1872** facilitates wireless connectivity between the system **1802** and the “outside world,” via a communications carrier or service provider. Transmissions to and from the radio **1872** are conducted under control of the operating system **1864**. In other words, communications received by the radio **1872** may be disseminated to the application
15 programs **1850** via the operating system **1864**, and vice versa.

[0079] According to an aspect, the visual indicator **1820** is used to provide visual notifications and/or an audio interface **1874** is used for producing audible notifications via the audio transducer **1825**. In the illustrated example, the visual indicator **1820** is a light emitting diode (LED) and the audio transducer **1825** is a speaker. These devices may be
20 directly coupled to the power supply **1870** so that when activated, they remain on for a duration dictated by the notification mechanism even though the processor **1860** 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 **1874** is used to provide audible signals to and receive audible
25 signals from the user. For example, in addition to being coupled to the audio transducer **1825**, the audio interface **1874** may also be coupled to a microphone to receive audible input, such as to facilitate a telephone conversation. According to an aspect, the system **1802** further includes a video interface **1876** that enables an operation of an on-board camera **1830** to record still images, video stream, and the like.

30 [0080] According to an aspect, a mobile computing device **1800** implementing the system **1802** has additional features or functionality. For example, the mobile computing device **1800** 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 18B** by the non-volatile storage area **1868**.

[0081] According to an aspect, data/information generated or captured by the mobile computing device **1800** and stored via the system **1802** are stored locally on the mobile computing device **1800**, 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 **1872** or via a wired connection between the mobile computing device **1800** and a separate computing device associated with the mobile computing device **1800**, 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 **1800** via the radio **1872** 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.

[0082] **FIGURE 19** 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 **1922**, a web portal **1924**, a mailbox service **1926**, an instant messaging store **1928**, or a social networking site **1930**. 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 **1920** provides the task list service **120** to clients **1905a,b,c**. As one example, the server **1920** is a web server providing the task list service **120** over the web. The server **1920** provides the task list service **120** over the web to clients **1905** through a network **1940**. By way of example, the client computing device is implemented and embodied in a personal computer **1905a**, a tablet computing device **1905b** or a mobile computing device **1905c** (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 **1916**.

[0083] 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.

[0084] 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
5 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
10 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 providing dynamic context for tasks to convey details and circumstances related to the task and supplementary information to augment the understanding of the task, comprising:
 - creating a task item;
 - observing a context of task item creation; and
 - associating the task item with the observed context in a relational graph.
2. The method of claim 1, further comprising:
 - querying the relational graph based on a queried context; and
 - returning one or more objects linked in the relational graph to the queried context.
3. The method of claim 1, wherein the observed context of task item creation includes:
 - documents open by a given user when the task item is created by the given user.
4. The method of claim 1, wherein the observed context of task item creation includes:
 - an event scheduled for a time in which the task item is created; and
 - identities of persons scheduled for the event.
5. The method of claim 4, wherein a node representing the task item in the relational graph is associated via edges with nodes representing the persons scheduled for the event in the relational graph.
6. The method of claim 5, wherein in response to receiving a query related to the node representing the task item in the relational graph, biographical details for the persons scheduled for the event are returned from the nodes representing the persons scheduled for the event.
7. The method of claim 1, further comprising:
 - observing a second context surrounding task item interaction; and
 - associating the task item with the second context in the relational graph.
8. The method of claim 7, wherein the second context is a new content item related to the task item, the new content item including at least one of:
 - a word processing file;
 - an email file;
 - a spreadsheet file;
 - a note taking file; and
 - a presentation file.

9. A method for providing dynamic context for tasks to convey details and circumstances related to the task and supplementary information to augment the understanding of the task, comprising:

receiving a graph query at a relational graph, the graph query requesting context for a task item;

spanning the relational graph from a node associated with the task item via contextual edges; and

returning objects linked to the node via the contextual edges according to the graph query.

10. The method of claim 9, wherein the node represents a given person to whom the task item is assigned.

11. The method of claim 9, wherein the node represents the task item.

12. The method of claim 9 wherein the contextual edges define a contextual relationship between the task item and contextual entities, the contextual entities including at least one of:

an event scheduled for the user to whom the task item is assigned at a time when the task item was created;

a person participating in the event; and

a document having focus at the time when the task item was created.

13. The method of claim 12, wherein the objects linked to the node via the contextual edges include contextual entities and other entities linked to the contextual entities.

14. The method of claim 13, wherein graph query specifies a number of nodes for the other entities to be within relative to the contextual entities in the relational graph.

15. A system including a processor and a computer readable storage device including processor executable instructions that when executed by the processor provide dynamic context for tasks to convey details and circumstances related to the task and supplementary information to augment the understanding of the task, comprising:

creating a task item;

observing a context of task item creation;

associating the task item with the observed context in a relational graph.

querying the relational graph based on a queried context; and

returning one or more entities linked in the relational graph to task item according to the queried context.

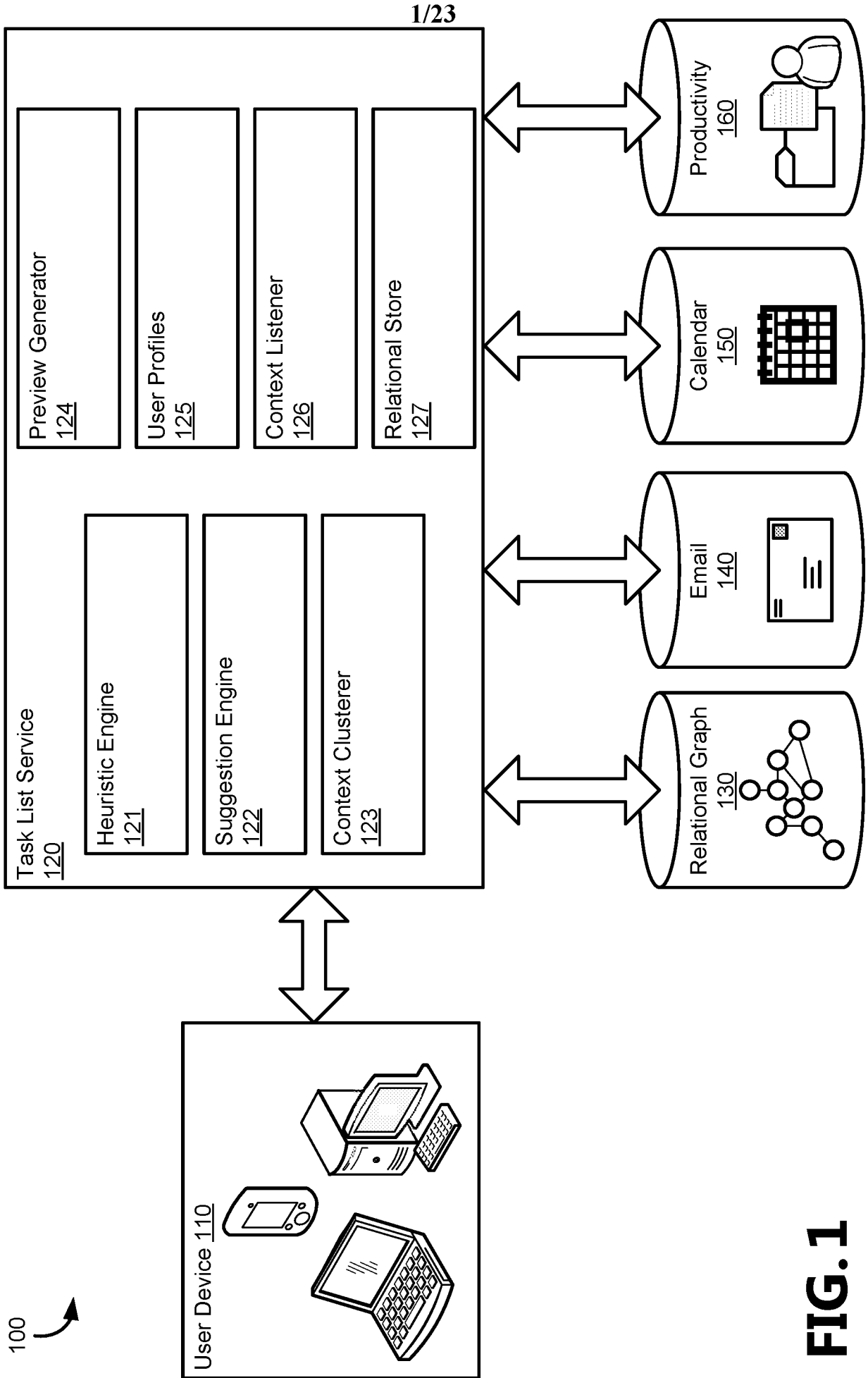


FIG. 1

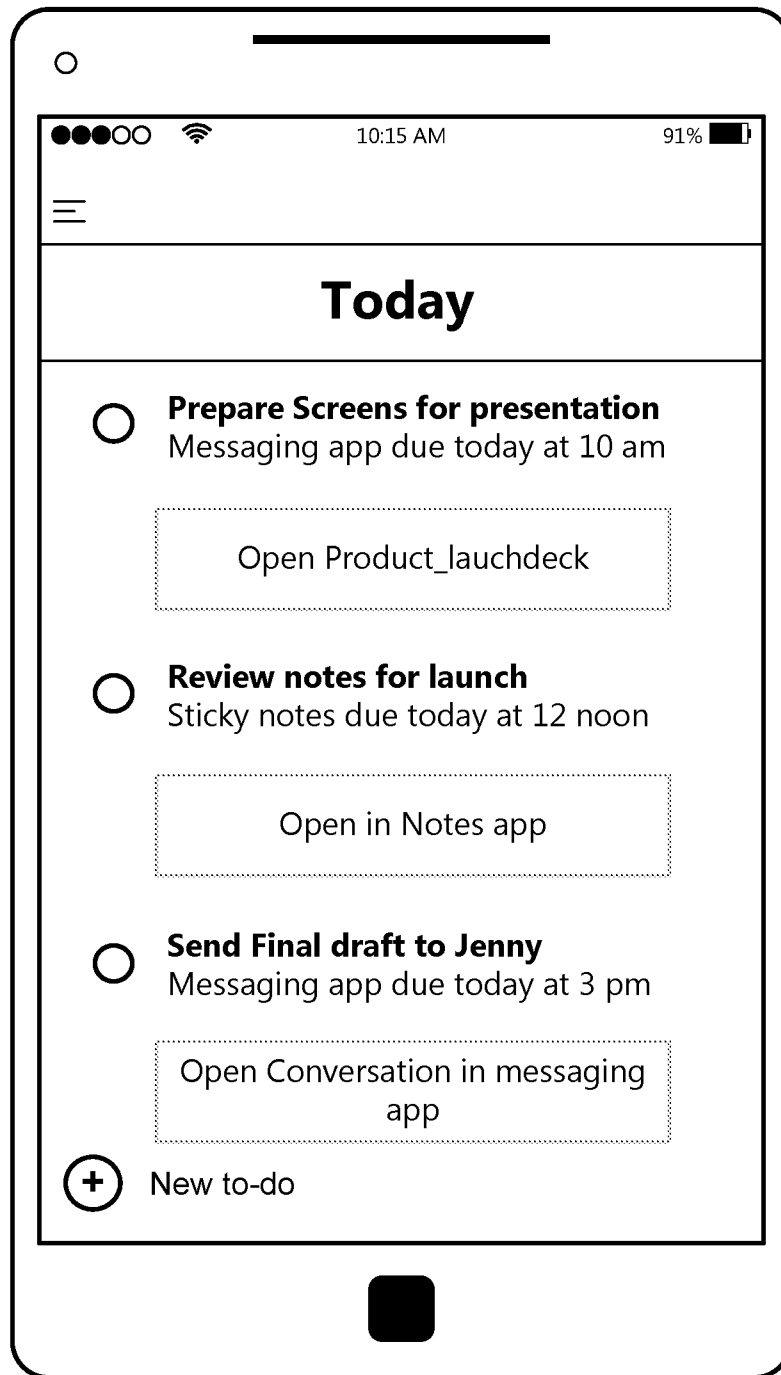


FIG. 2A

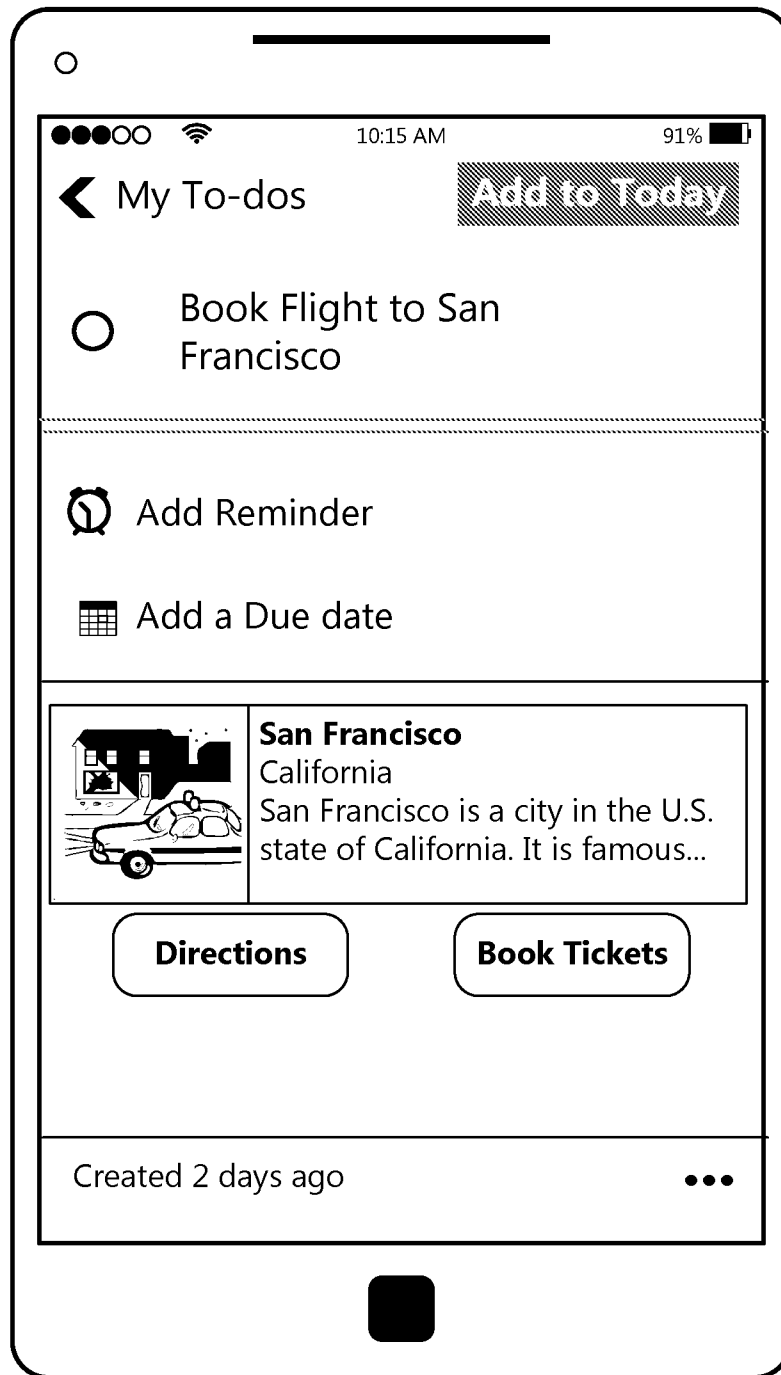


FIG. 2B

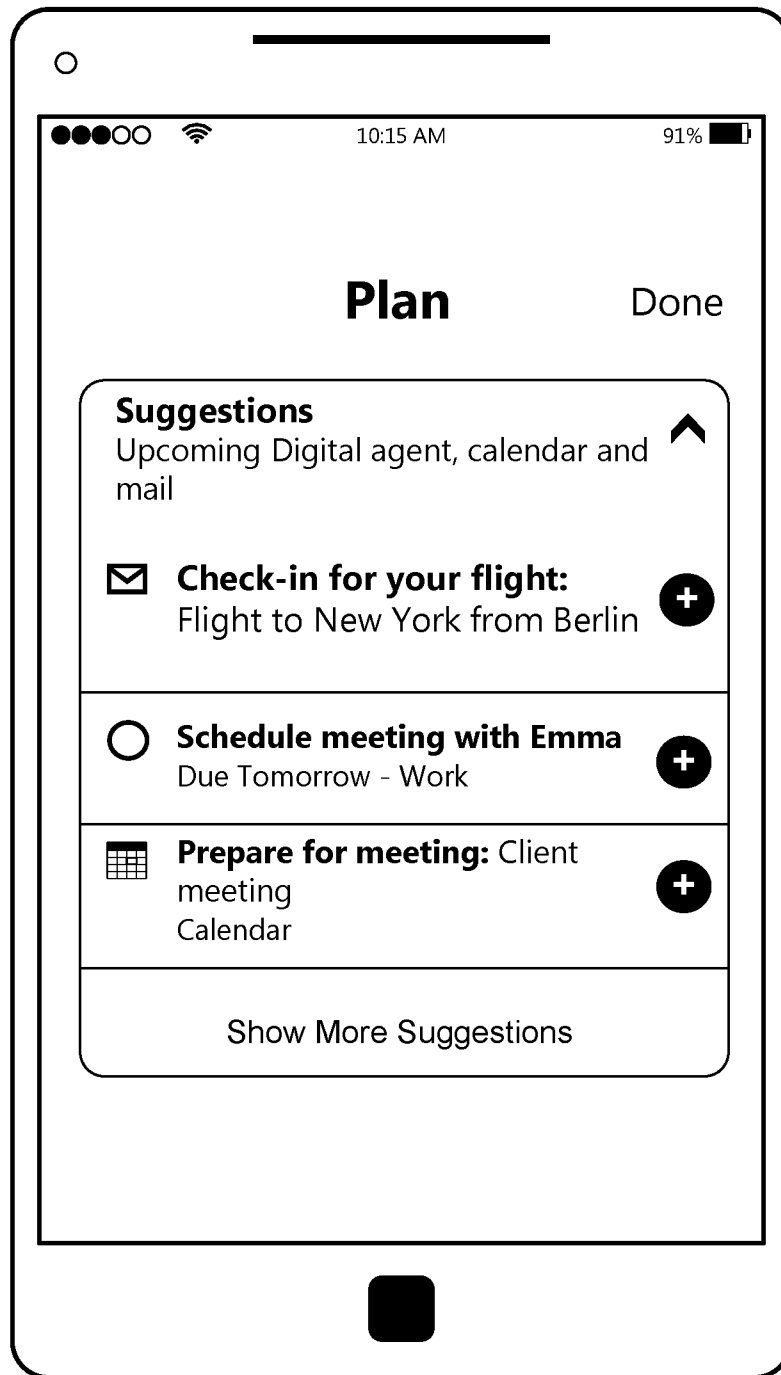


FIG. 3

5/23

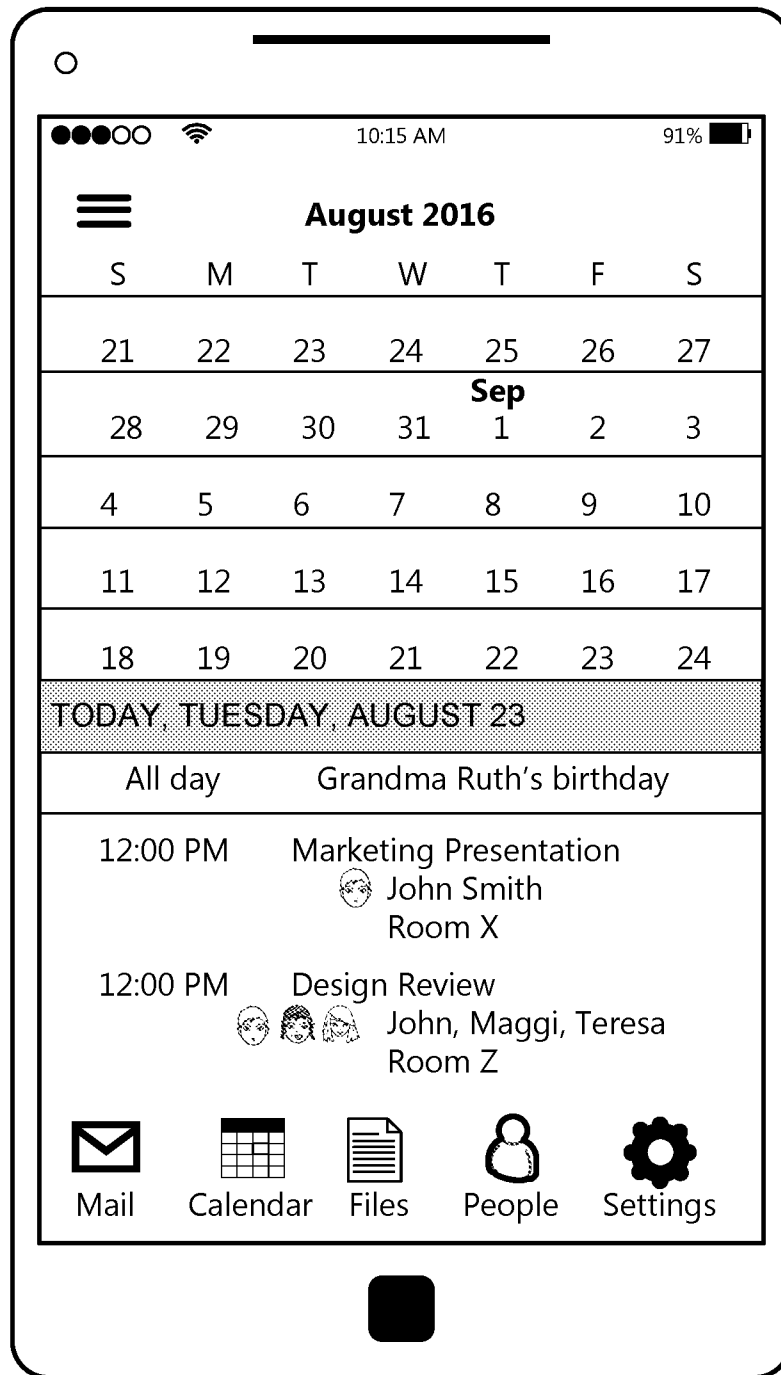


FIG. 4A

6/23

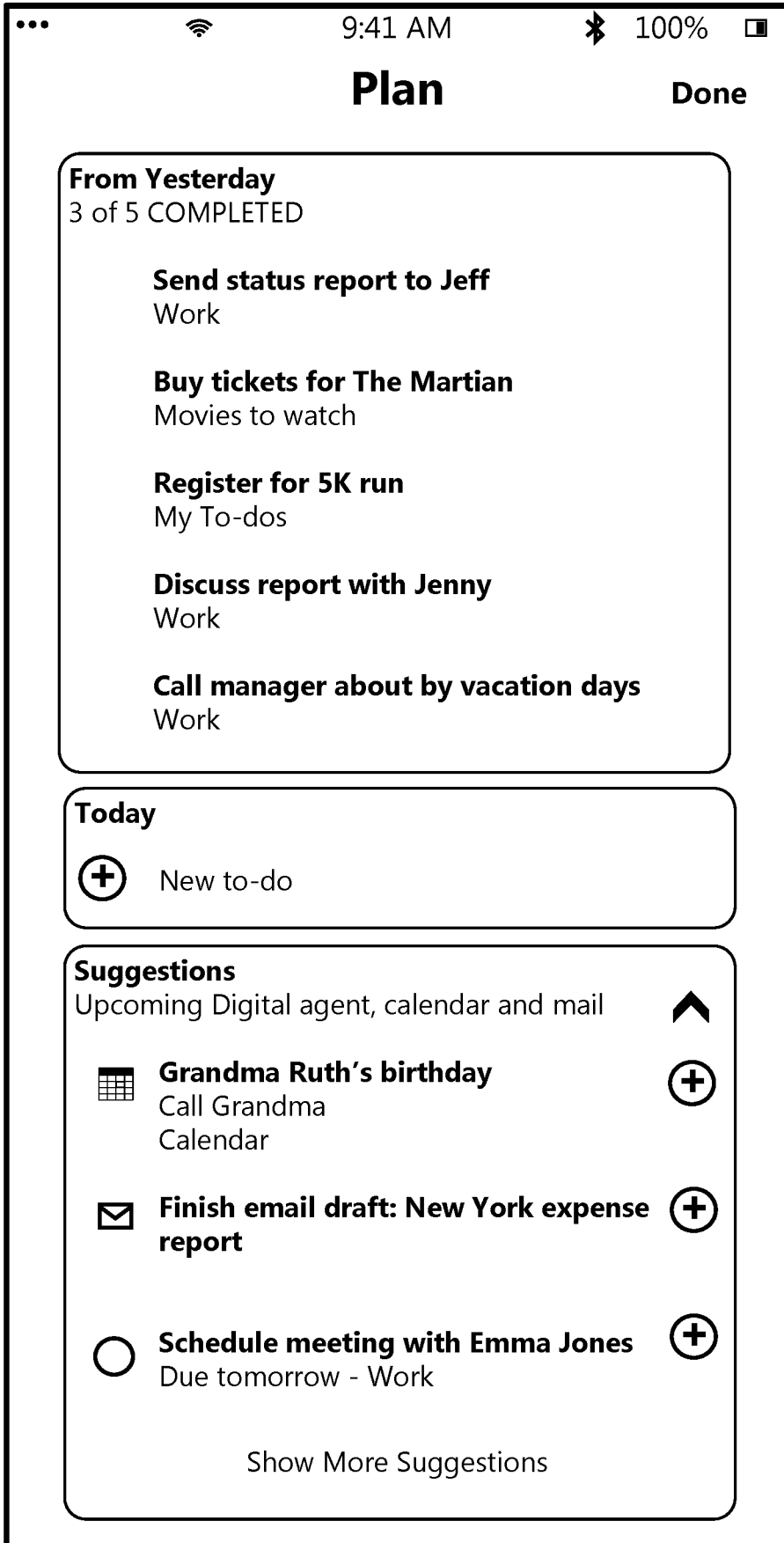


FIG. 4B

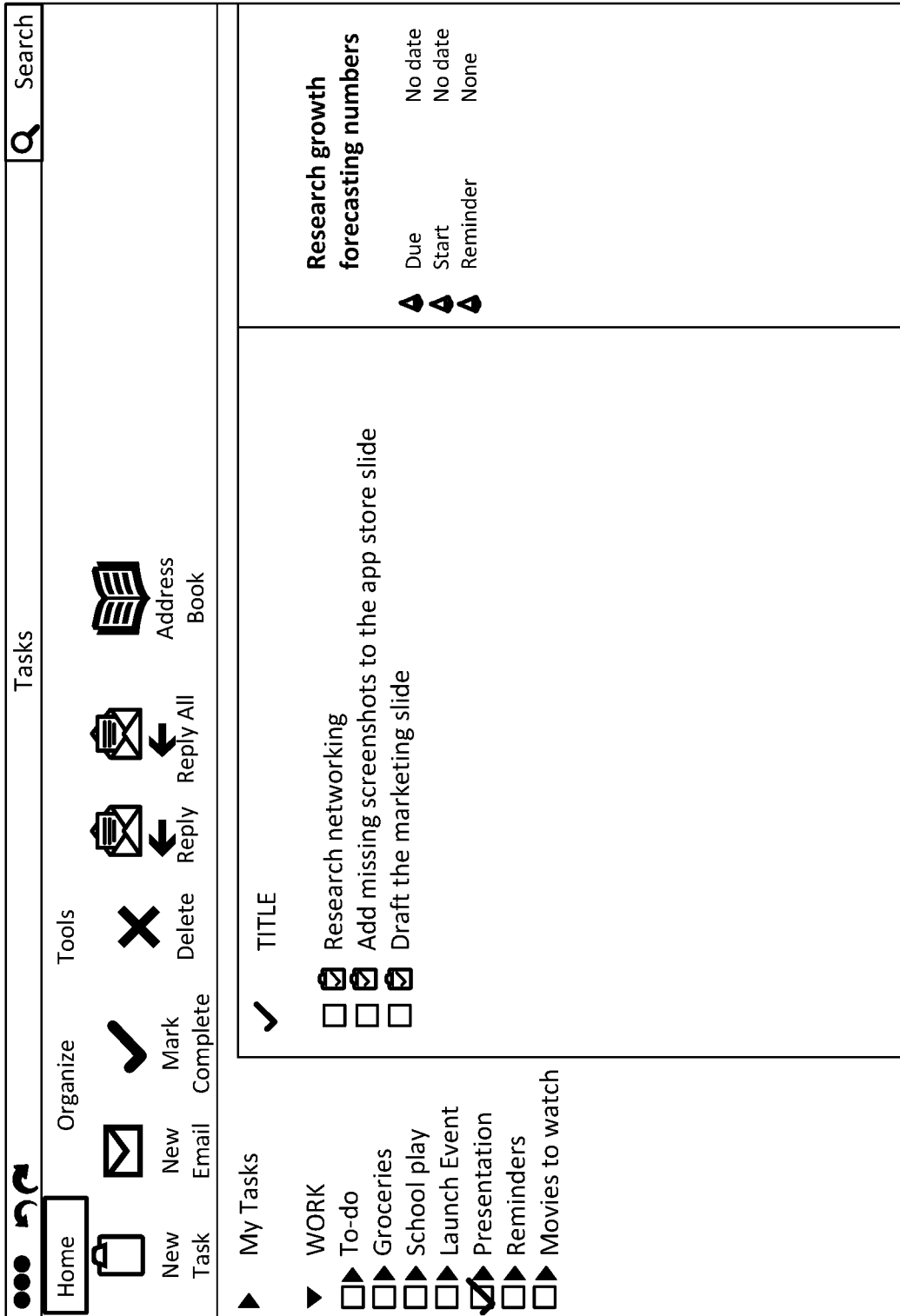


FIG. 5

8/23

<p>INBOX</p> <p> New mail</p> <p> Accounts</p> <p>John's Email 137</p> <p>John's Email 7</p> <p>Photography 2740</p> <p>Mail</p> <p> FOLDERS</p> <p>Inbox 2740</p> <p>Drafts 1</p> <p>Sent mail</p> <p>All Mail</p> <p>More</p>		<p>Search</p> <p>Inbox</p> <p>Sports channel</p> <p>The XYZ Show Newsletter for 3.23.13</p> <p>The XYZ Show Newsletter for 3.23.13 2:29 pm</p> <p>Rob Will, Environment</p> <p>The next solar panel</p> <p>Dear Cathy, this could be the next solar panel</p> <p>Kats, Jessica</p> <p>RE: The Tradition</p> <p>Excellent- I agree. Could you register? If not, I'll</p> <p>Stella from T Company</p> <p>Get productive</p> <p>#ReadySetGoal Bring some innovation into your</p> <p>Laying it all out special offers</p> <p>Crochet Hooks</p> <p>Allcrochetpatterns.com is happy to announce</p> <p>Bed Bath Store</p> <p>Closeout bedsheets 69.99 20% Off offer</p> <p>Value Bedding shop now view as web page</p> <p>Food Critics</p> <p>Reached 100 followers</p> <p>Hello food blogger, you have reached 100 foll</p>	<p> Reply Reply all Forward Task Delete </p> <p> Joanna</p> <p>RE: The Tradition</p> <p>To: Joanna, Mark, Jim</p> <p>Excellent- I agree. Could you register? If not, I'll try to remember to do so tonight</p> <p>Sent from Mail</p> <p>From: Cathy</p> <p>Sent: Tuesday, March 22, 2016, 11:51 am</p> <p>To: Jeff, Cathy</p> <p>Subject: The Tradition</p> <p>Hi Friends,</p> <p>Looks like Volunteer registration is open for the May Sale! We'd like you to volunteer both nights</p> <p>-Cathy</p>
---	--	--	--

FIG. 6

9/23

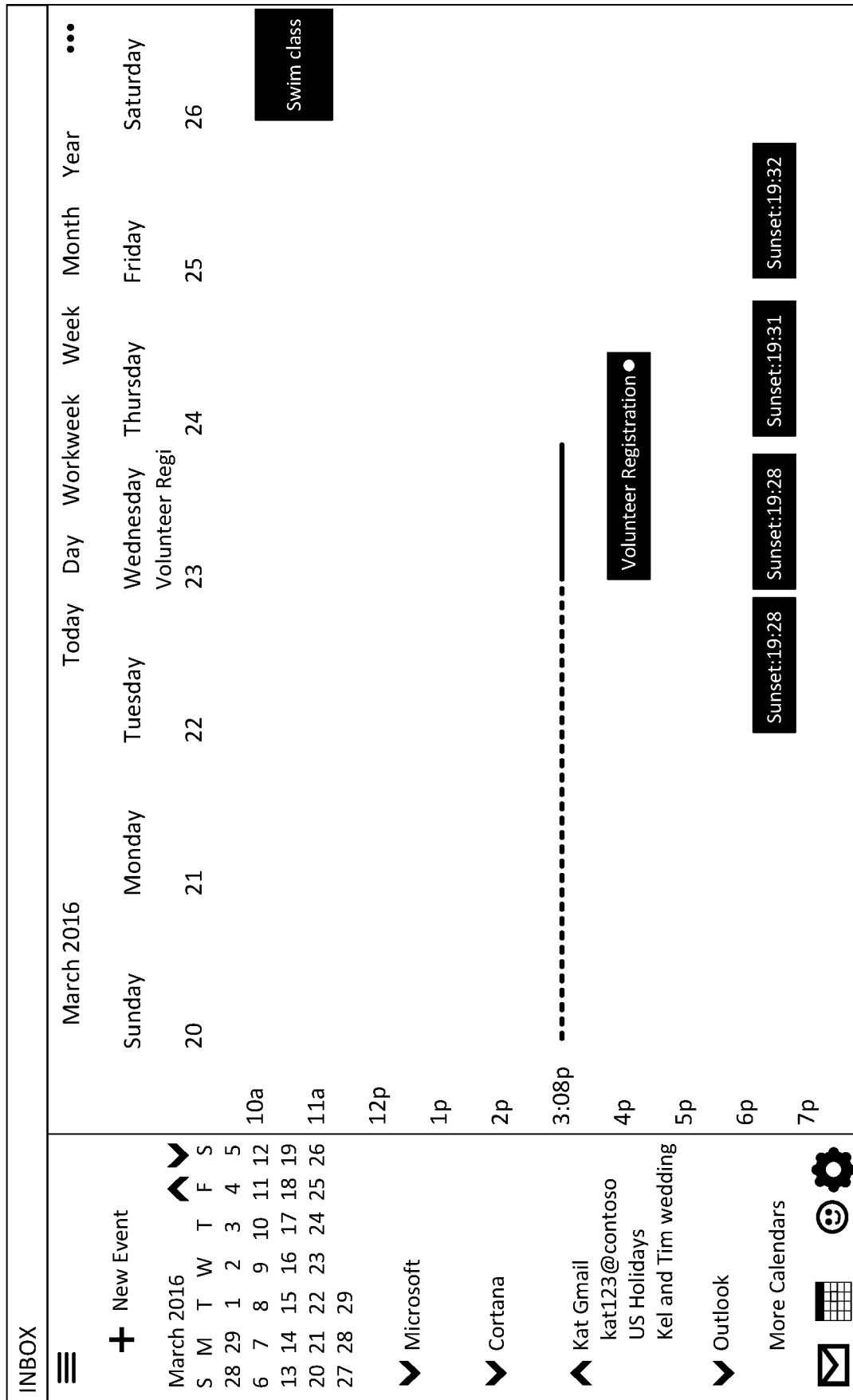


FIG. 7

Presentation

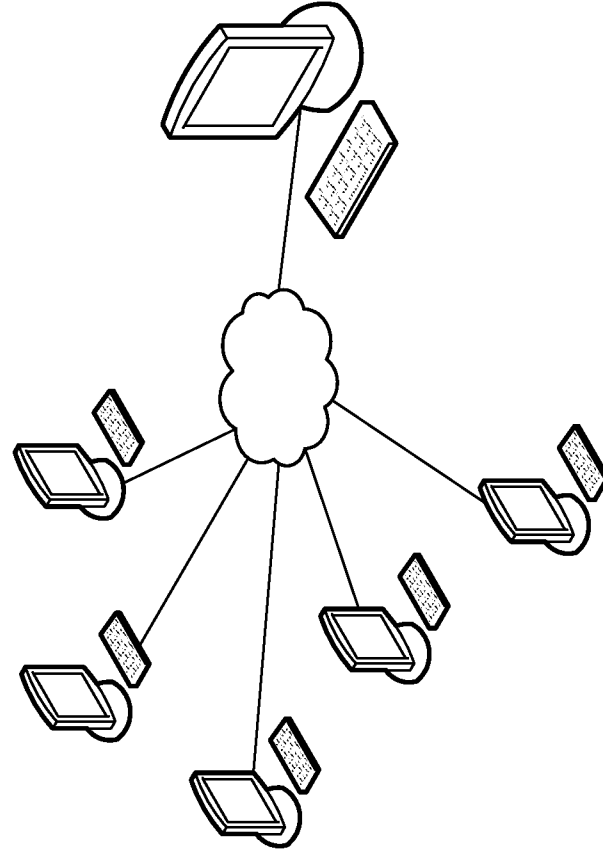
File Home 12 Cambria Slides Outline

Insert Design Transitions Animations Slide Show Review View

B I U ABC A Abc

Heading 1

NETWORKS



Tasks X


- Research Networks
- Add Missing screenshots to the slide
- Draft the Marketing slide
- Add a new to-do

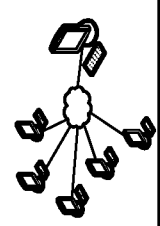
Congrats on adding new to-dos!

Do you know you can see Presentation application tasks in your task application?

View your To-Dos

1 COMPUTERS

2 What is a computer? 

3 NETWORKS 


4 The End Questions? 

FIG. 8

Task application
— □ ×

☰

Movies to watch

2 of 6 completed

- Jason Bourne
- The Martian
Due in 2 days
- Zeetopia
- The Jungle Book
- Batman vs. Superman

↕ ↕

Sort

⋮ More

Add to Today


○ **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

🗑️

Hide

+ New to-do

FIG. 9

12/23

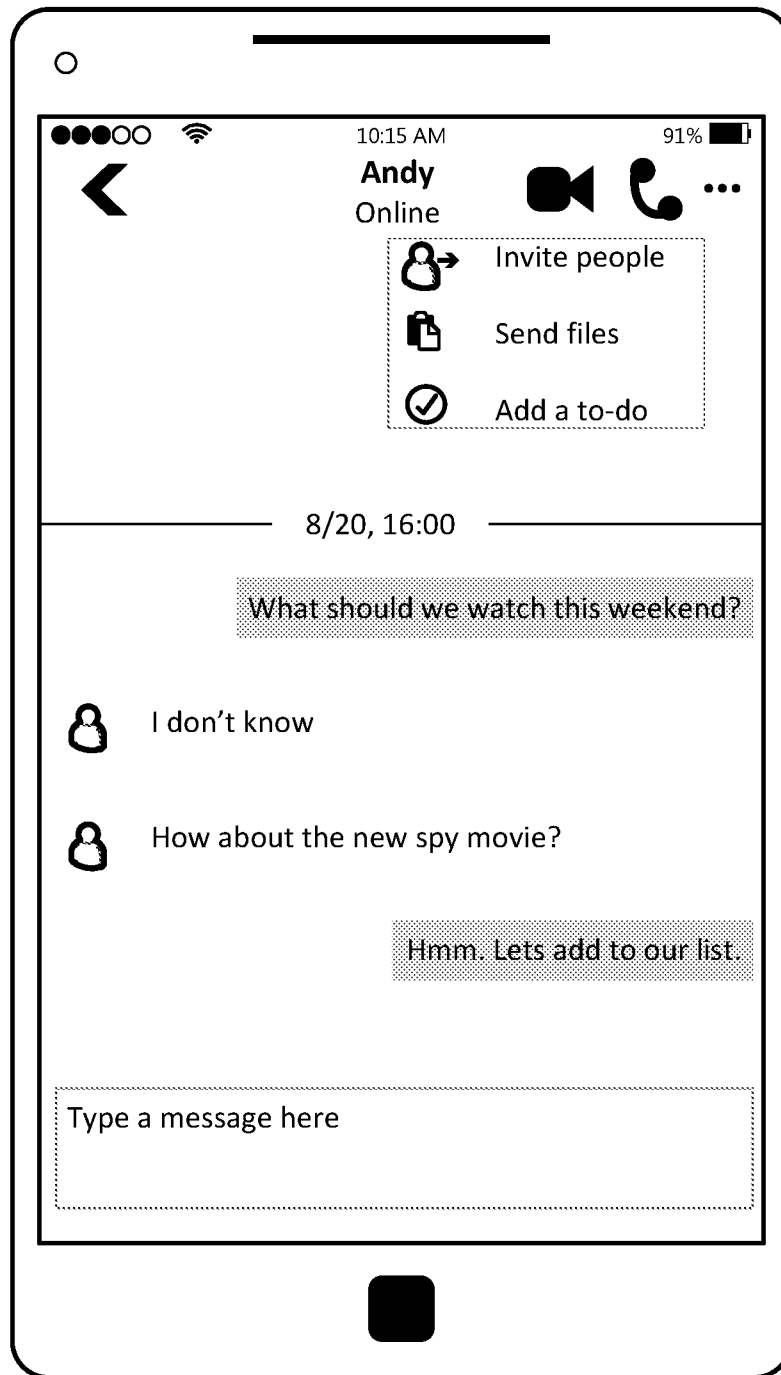


FIG. 10



FIG. 11

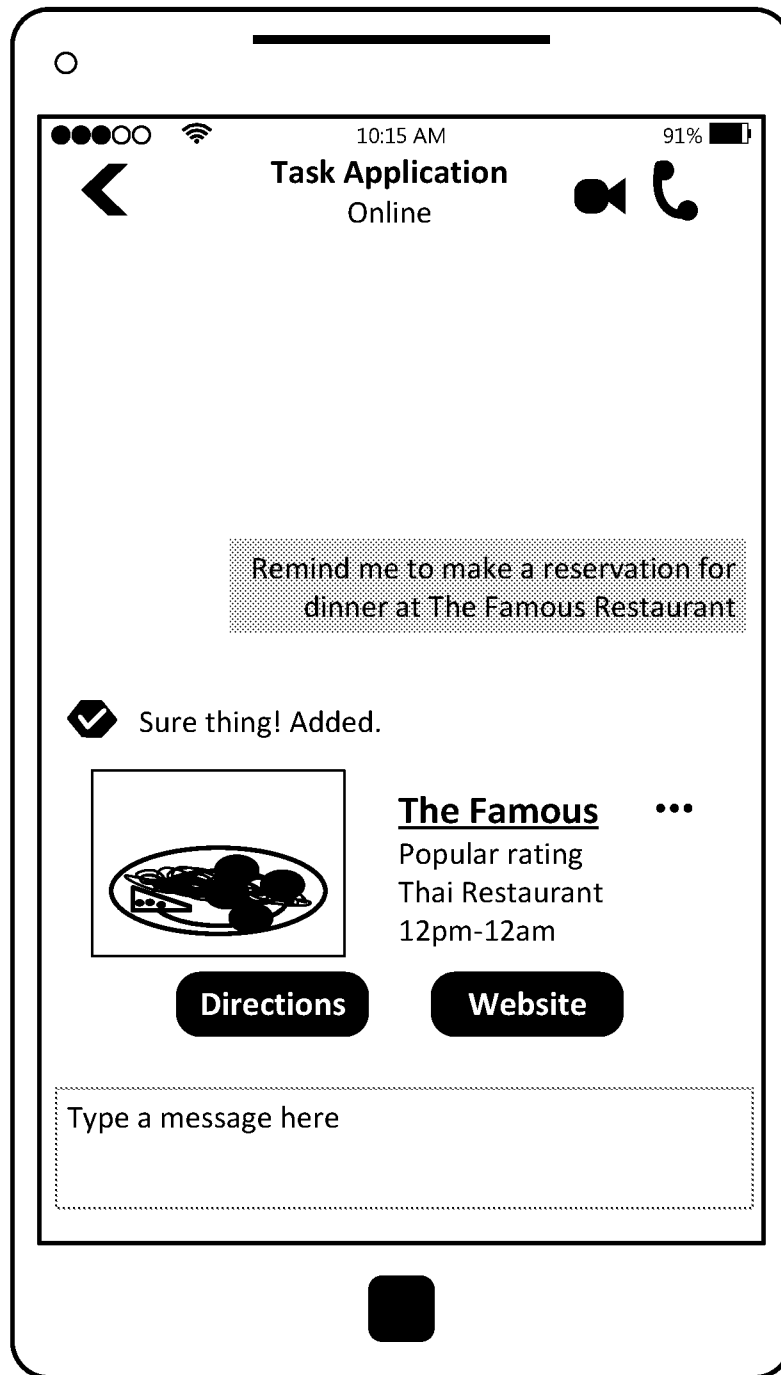


FIG. 12

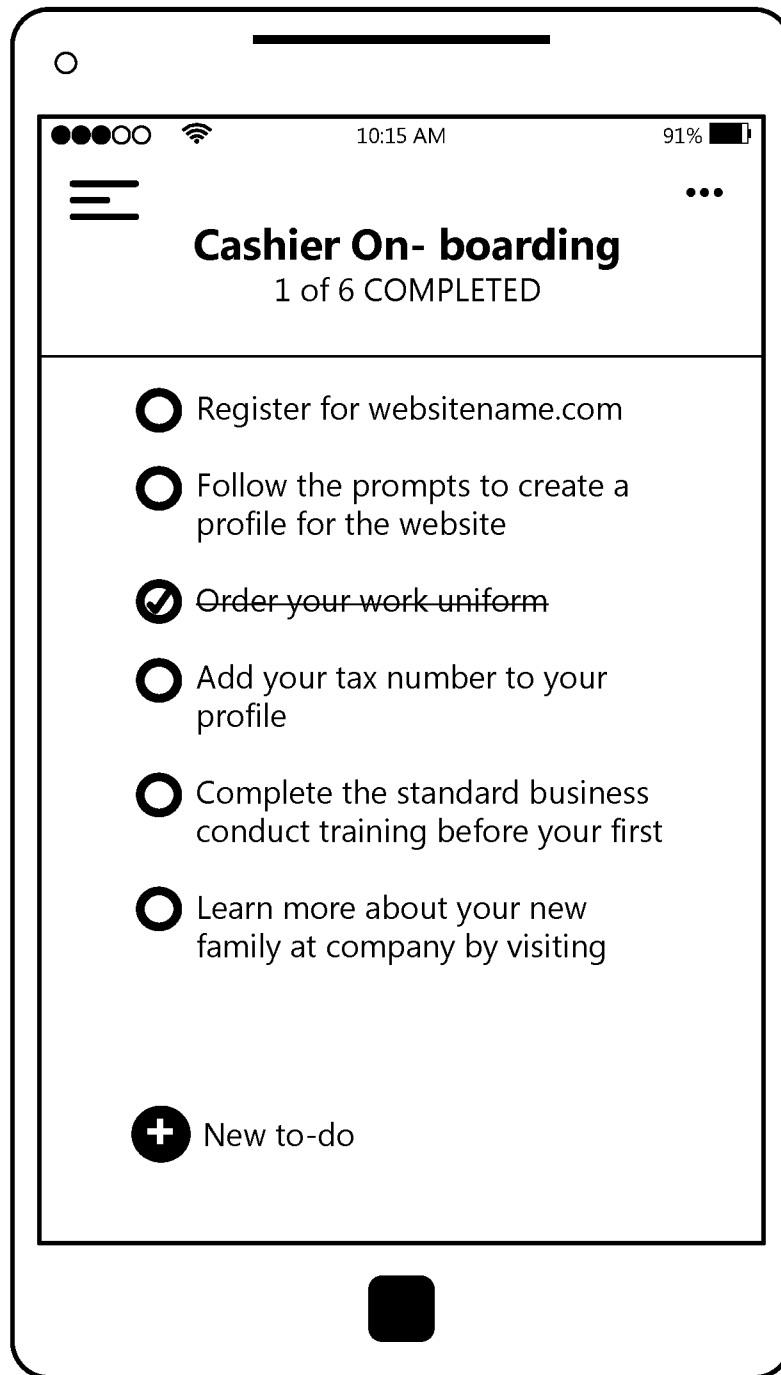


FIG. 13

Presentation App

Tasks

Today

- Add consistent title with last report *presentation today for John*
- Adapt Formatting *Today*

Upcoming

- Add consistent title with last report *presentation today for John*
- Adapt Formatting *Today*
- Contextual app meet *Overdue since last week*

Open other Presentations

— □ X
 John Smith
 johnsmith@company.com
 Switch account

Take a tour

Make Interactive Videos

GALLERY

PARCEL

WOOD TYPE

ION

Quotable

BERLIN

CELESTIAL

SAVON

CROP

CIRCUIT

DIVIDEND

Frame

Organic

FIG. 14A

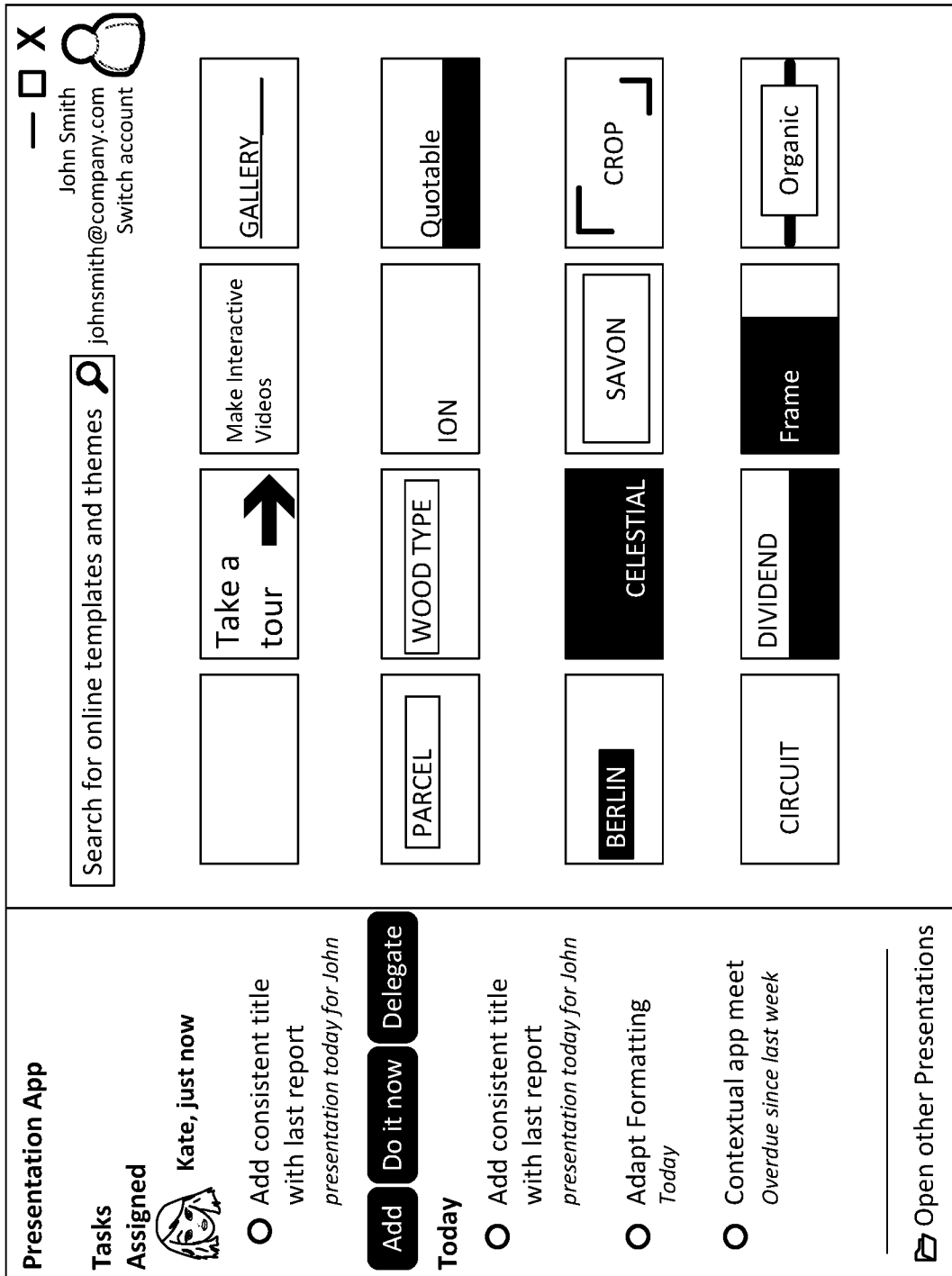


FIG. 14B

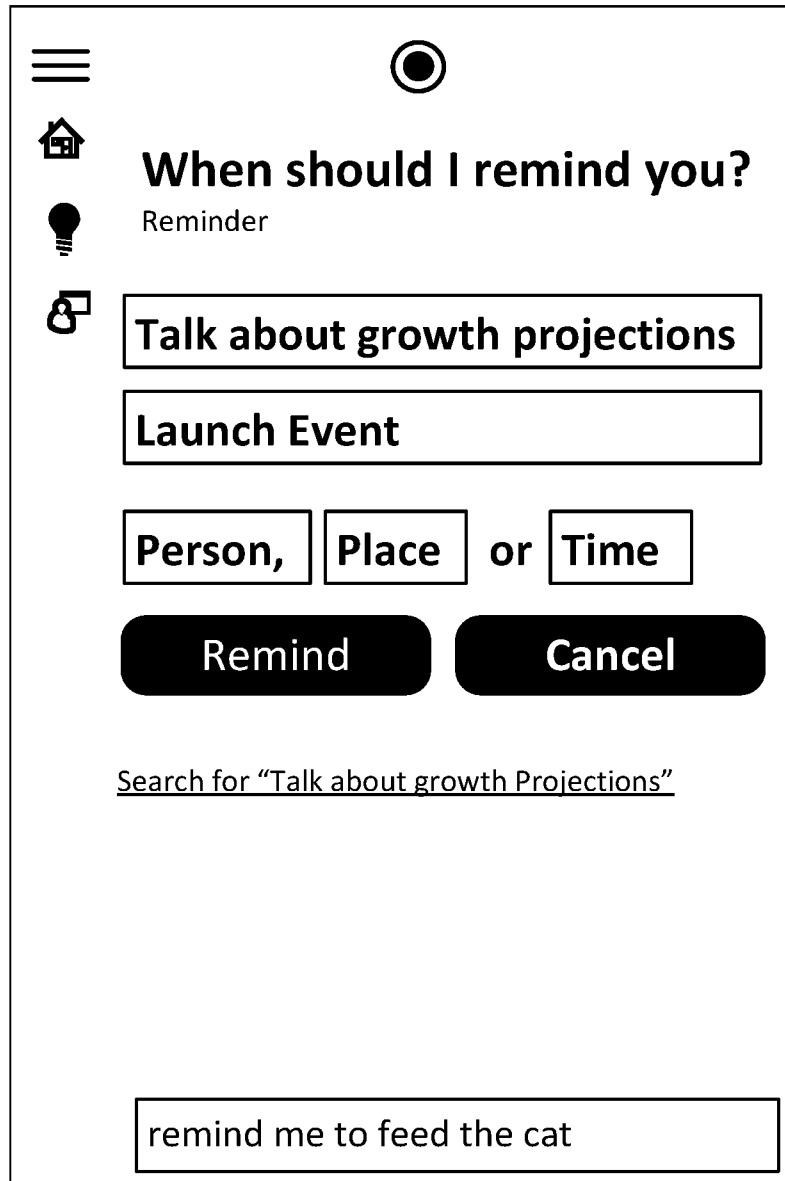


FIG. 15

1600



19/23

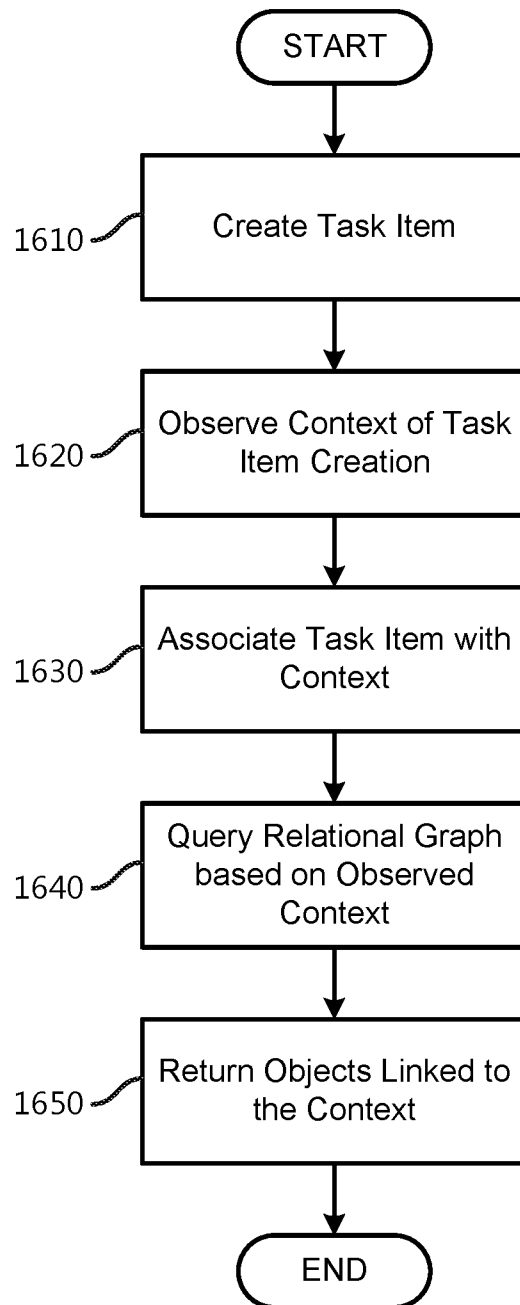


FIG. 16

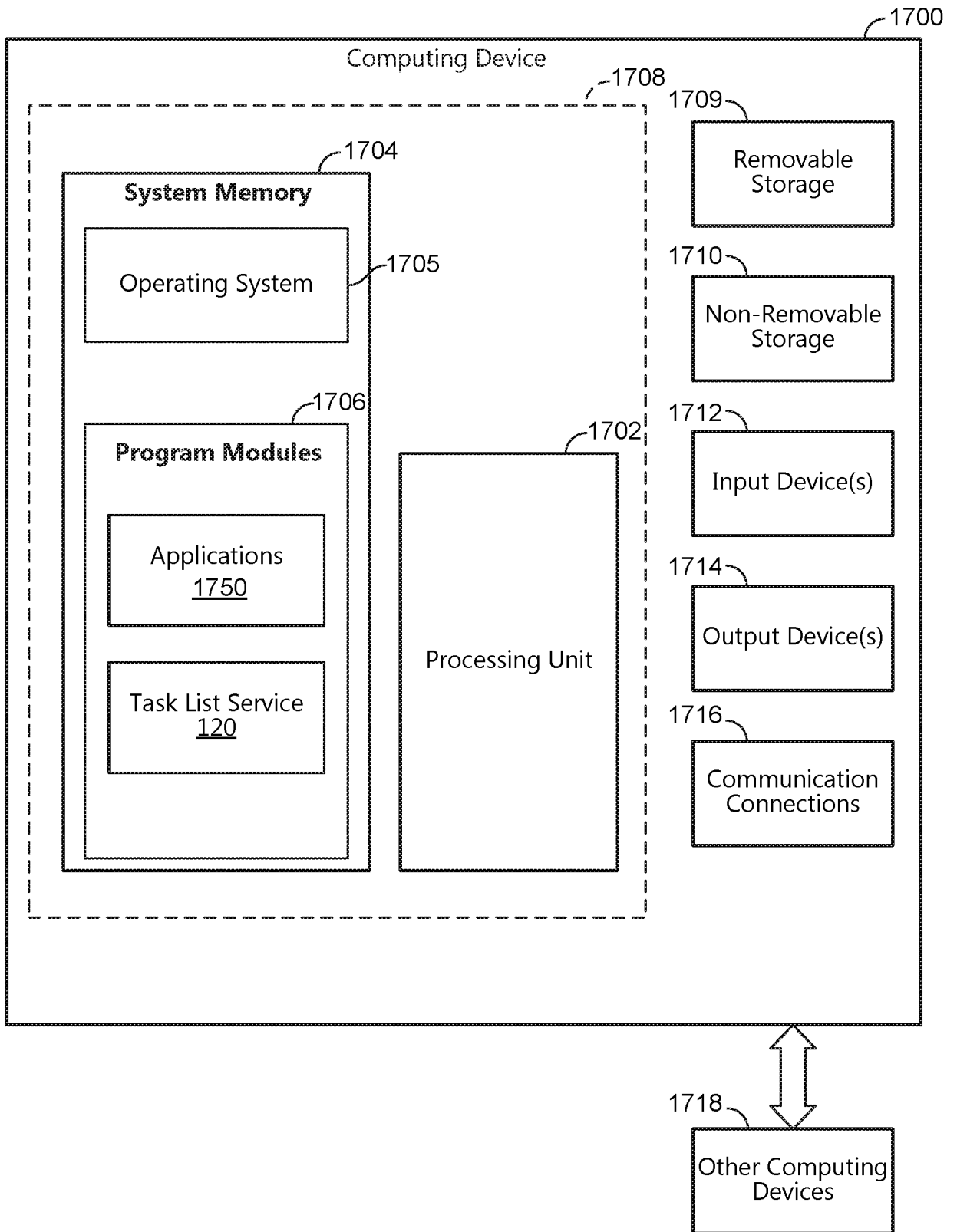
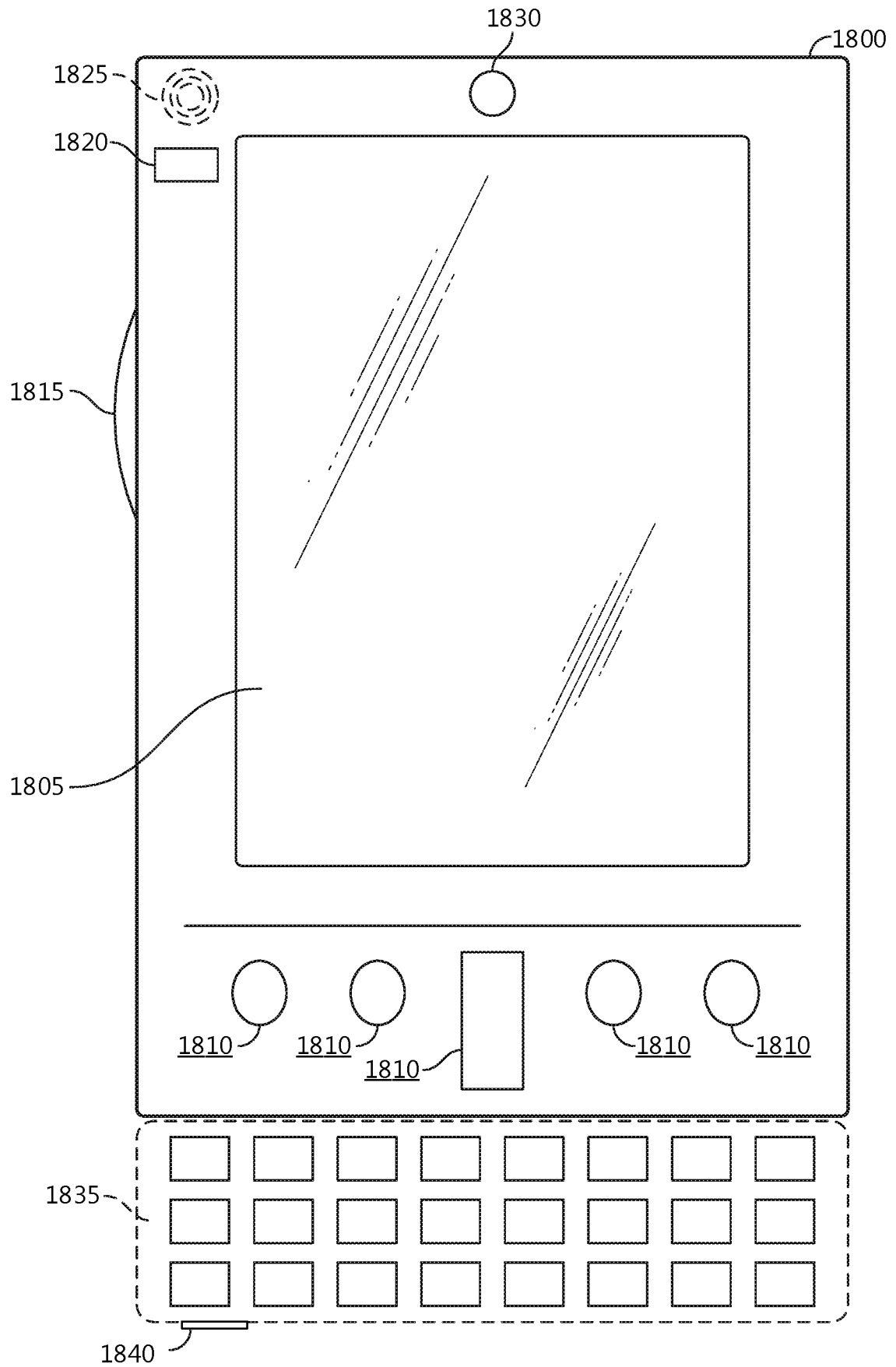


FIG. 17

21/23



MOBILE COMPUTING DEVICE

FIG. 18A

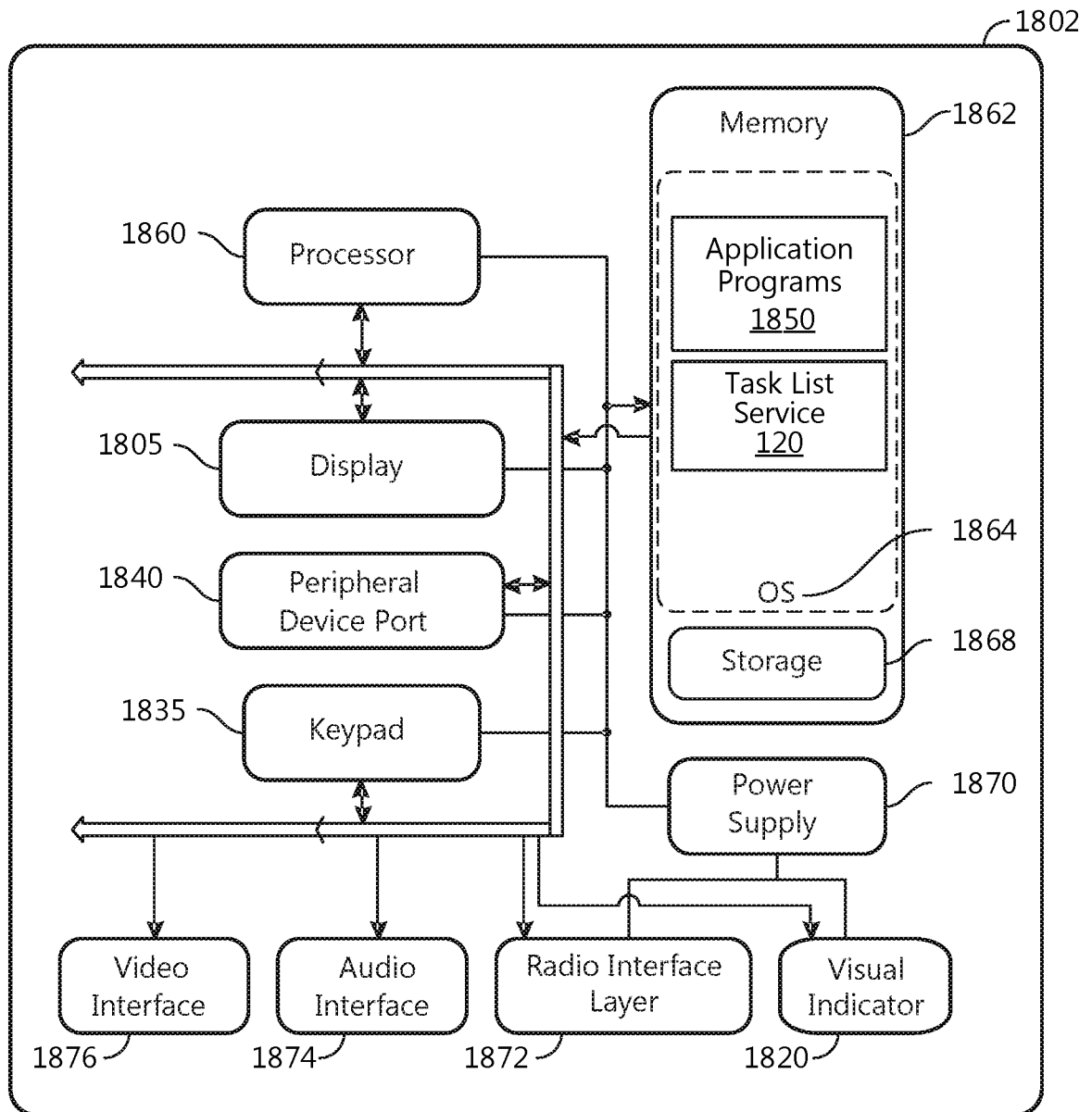


FIG. 18B

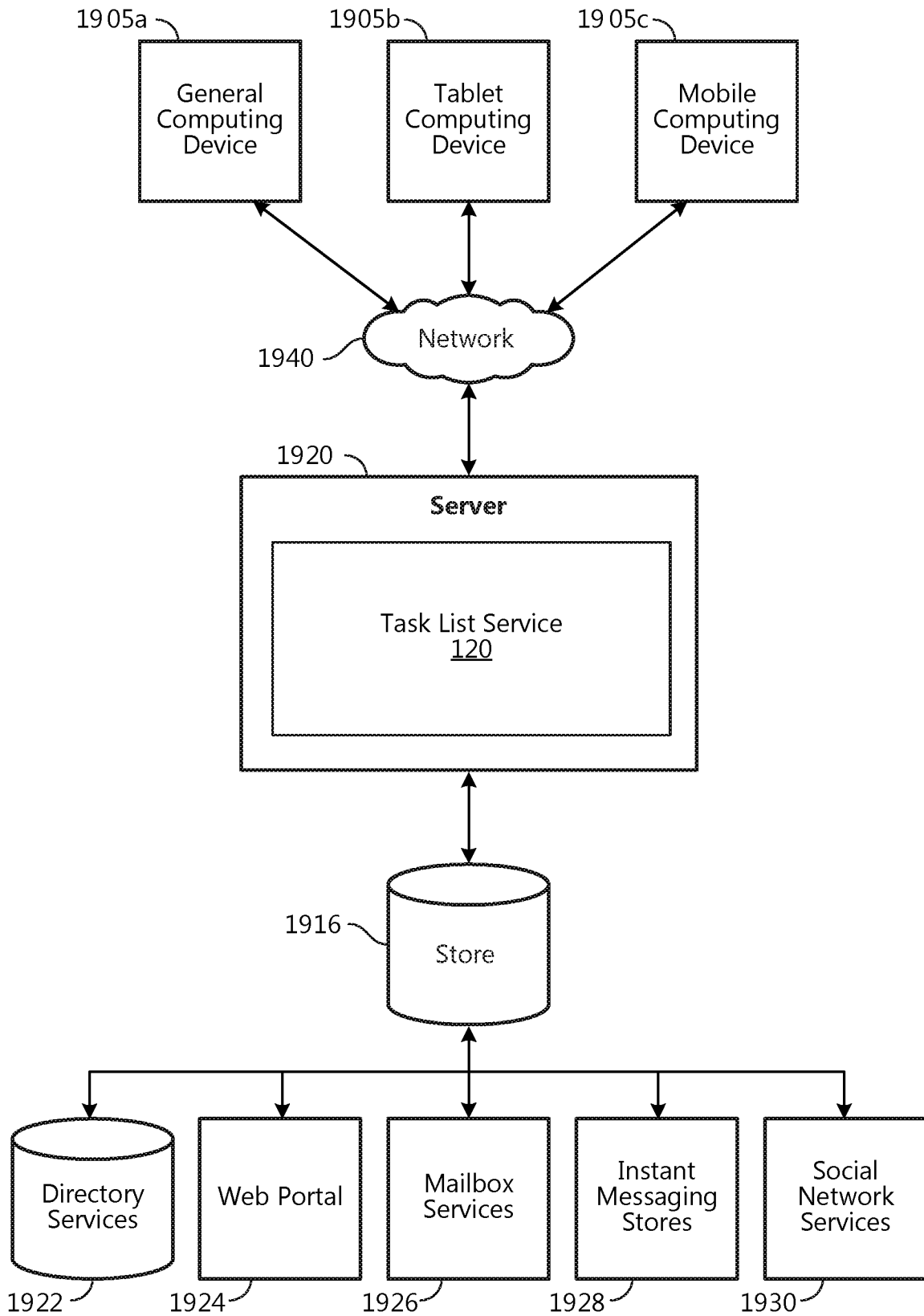


FIG. 19

INTERNATIONAL SEARCH REPORT

International application No PCT/US2017/059633

A. CLASSIFICATION OF SUBJECT MATTER INV. G06Q10/06 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 2015/242091 A1 (LU QI [US] ET AL) 27 August 2015 (2015-08-27) abstract, paragraphs 4, 5, 13, 20, 29-59; figures -----	1-15		
X	US 2010/125548 A1 (ANZAI TAKESHI [JP] ET AL) 20 May 2010 (2010-05-20) paragraphs [0059] - [0067]; figures -----	1-15		
X	US 2016/086116 A1 (RAO SUPRIYA [IN] ET AL) 24 March 2016 (2016-03-24) paragraphs [0013] - [0051]; figures -----	1-15		
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.				
* Special categories of cited documents : <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none; vertical-align: top;"> "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 </td> <td style="width: 50%; border: none; vertical-align: top;"> "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 </td> </tr> </table>			"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
"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			
15 December 2017	03/01/2018			
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Blackley, William			

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2017/059633

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015242091 A1	27-08-2015	US 2015242091 A1 WO 2015127404 A1	27-08-2015 27-08-2015
US 2010125548 A1	20-05-2010	JP 4990258 B2 JP 2010117967 A US 2010125548 A1	01-08-2012 27-05-2010 20-05-2010
US 2016086116 A1	24-03-2016	NONE	