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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 July 2017 (06.07.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/113624 A1

(51) International Patent Classification:
G06F 3/0488 (2013.01)

(21) International Application Number:
PCT/CN2016/085678

(22) International Filing Date:
14 June 2016 (14.06.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN2015/099783
30 December 2015 (30.12.2015) CN
PCT/CN2016/079883 21 April 2016 (21.04.2016) CN

(71) Applicant: **BEIJING CHAOZHUO TECHNOLOGY CO., LTD.** [CN/CN]; Room 231808, 18th Floor, Building C, Wangjing SOHO Building T2, East Futong, Main Street, Chaoyang, Beijing 100102 (CN).

(72) Inventors: **XIE, Zhenyu**; Room 231808, 18th Floor, Building C, Wangjing SOHO Building T2, East Futong, Main Street, Chaoyang, Beijing 100102 (CN). **JI, Xi**; Room 231808, 18th Floor, Building C, Wangjing SOHO Building T2, East Futong, Main Street, Chaoyang, Beijing 100102 (CN).

(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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, 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).

Published:

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

(54) Title: SYSTEM AND METHOD FOR OPERATING SYSTEM OF MOBILE DEVICE

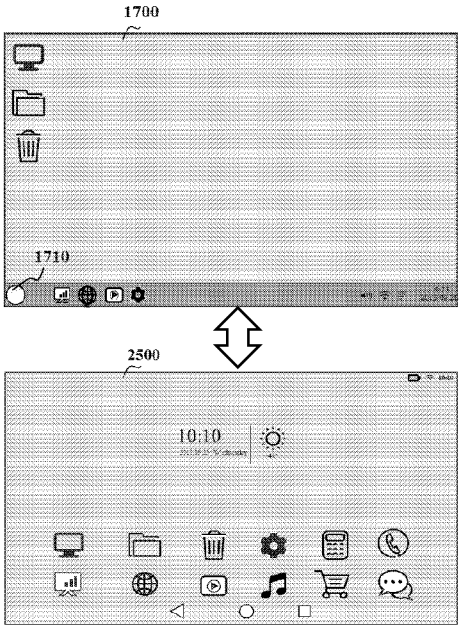

FIG. 25

(57) Abstract: In accordance with various aspects of the present disclosure, a method for providing a mobile operating system is disclosed. The method includes: displaying a desktop interface on a screen of a mobile device; and displaying a first application window and a second application window on the desktop interface, wherein the first application window corresponds to a first application running on the mobile device, wherein the second application window corresponds to a second application running on the mobile device, and wherein at least a portion of the first application window does not overlap with the second application window.

WO 2017/113624 A1

SYSTEM AND METHOD FOR OPERATING SYSTEM OF MOBILE DEVICE

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims priority of PCT application No. PCT/CN2015/099783 filed on December 30, 2015, PCT application No. PCT/CN2016/079883 filed on April 21, 2016, each of which is hereby incorporated by reference.

TECHNICAL FIELD

[0002] This application generally relates to the field of computing systems, and more particularly to a mobile device and a mobile operating system supporting multi-window features.

BACKGROUND

[0003] Mobile devices, such as mobile phones and tablet computers, are becoming increasingly popular. Typically a mobile device employs a touch screen that allows user input through detecting the presence and location of a touch in a display area. A conventional operating system (OS) of such a mobile device differs from the Windows™ OS and provides a user with various functions based on the touch-screen feature of the mobile device. For example, unlike the Windows™ OS, the Android™ operating system and other conventional mobile operating systems do not provide a user with start menu, task bar, multi-window display, and other desirable managerial functions. Instead, conventional mobile operating systems display user interfaces of an application running on the mobile phone in a full-screen mode. In addition, the conventional mobile operating systems cannot simultaneously display user interfaces of multiple applications running on a mobile device on a screen of the mobile device. As such, conventional mobile operating systems do not provide a user with the ability to simultaneously interact with multiple applications running on a mobile device. This can, for example, create a frustrating experience for users of conventional mobile devices.

[0004] Accordingly, it is desirable to provide a mobile operating system with enhanced features.

SUMMARY

[0005] The following is a simplified summary of the disclosure in order to provide a basic understanding of some aspects of the disclosure. This summary is not an extensive overview of the disclosure. It is intended to neither identify key or critical elements of the disclosure, nor delineate any scope of the particular implementations of the disclosure or any scope of the claims. Its sole purpose is to present some concepts of the disclosure in a simplified form as a prelude to the more detailed description that is presented later.

[0006] In some embodiments, a method for providing a mobile operating system is disclosed. The method includes: displaying a desktop interface on a screen of a mobile device; and displaying a first application window and a second application window on the desktop interface, wherein the first application window corresponds to a first application running on the mobile device, wherein the second application window corresponds to a second application running on the mobile device, and wherein at least a portion of the first application window does not overlap with the second application window.

[0007] In some embodiments, the mobile operating system is based on the Android™ operating system.

[0008] In some embodiments, the method further includes displaying a task bar on the desktop interface, wherein the taskbar comprises a first visual representation of the first application and a second visual representation of the second application.

[0009] In some embodiments, the desktop interface includes a start menu button for displaying a start menu.

[0010] In some embodiments, displaying the first application window and the second application window on the desktop interface further includes: querying an application configuration list for a first size and a first coordinate; and displaying the first application window based on the first size and the first coordinate.

[0011] In some embodiments, displaying the first application window and the second application window on the desktop interface further includes: querying the application configuration list for a second size and a second coordinate; and displaying the second application window based on the second size and the second coordinate.

[0012] In some embodiments, the second size corresponds to a type of the second application.

[0013] In some embodiments, the type of the second application is one of a phone application and a tablet application.

[0014] In some embodiments, at least a portion of the first application window overlaps with the second application window.

[0015] In some embodiments, the method further includes: detecting a user interaction with at least one of the first application window and the second application window; and rearranging the first application window and the second application window based on the user interaction, wherein the rearranged first application window is partially overlapped with the second application window.

[0016] In some embodiments, the method further includes: displaying a visual representation of an object in the first application window; receiving a user input that moves the object from the first application window to the second application window; and displaying the visual

representation of the object in the second application window. In some embodiments the user input corresponds to a drag and drop gesture.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] Various objects, features, and advantages of the disclosed subject matter can be more fully appreciated with reference to the following detailed description of the disclosed subject matter when considered in connection with the following drawings, in which like reference numerals identify like elements.

[0018] **FIG. 1** is a block diagram illustrating an example of a mobile device in accordance with some embodiments of the present disclosure.

[0019] **FIG. 2** is a flowchart illustrating a process of input execution by a mobile device in accordance with some embodiments of the present disclosure.

[0020] **FIG. 3** is a block diagram of a mobile device in accordance with some embodiments of the present disclosure.

[0021] **FIG. 4** is a block diagram illustrating an example of a mobile operating system (OS) that may be employed in a mobile device in accordance with some embodiments of the present disclosure.

[0022] **FIG. 5** is a block diagram illustrating an example of a kernel of a mobile operating system in accordance with some embodiments of the present disclosure.

[0023] **FIG. 6** is a block diagram illustrating a mobile OS and its interactions with a kernel and hardware in accordance with some embodiments of the present disclosure.

[0024] **FIG. 7** illustrates a multi-window framework architecture in accordance with some embodiments of the present disclosure.

[0025] **FIG. 8** is a block diagram illustrating an application configuration list and an application type list in accordance with some embodiments of the present disclosure.

[0026] **FIG. 9A** is a flow diagram illustrating an example of a process for presenting multiple application windows on a mobile device in accordance with some embodiments of the present disclosure.

[0027] **FIG. 9B** is a flowchart illustrating an example of a process of multi-window application management in accordance with some embodiments of the present disclosure.

[0028] **FIG. 10** is a flowchart illustrating a process for arranging application windows for presentation on a mobile device in accordance with some embodiments of the present disclosure.

[0029] **FIG. 11** illustrates an example of a desktop interface in accordance with some embodiments of the present disclosure.

[0030] **FIG. 12** is a flowchart illustrating a method for arranging application windows in accordance with some embodiments of the present disclosure.

[0031] **FIG. 13** is a diagram illustrating a mechanism for running an application on various mobile devices in accordance with some embodiments of the present disclosure.

[0032] **FIG. 14** is a flowchart illustrating a method for presenting application windows of an application on various mobile devices in accordance with some embodiments of the present disclosure.

[0033] **FIG. 15** illustrates a mechanism for moving an object across multiple application windows in accordance with some embodiments of the present disclosure.

[0034] **FIG. 16** is a flowchart illustrating a process for moving an object across multiple application windows in accordance with some embodiments of the present disclosure.

[0035] **FIG. 17** is a user interface that may be employed by a mobile device in accordance with some embodiments of the present disclosure.

[0036] **FIG. 18** illustrates a start menu in accordance with one embodiment of the present disclosure.

[0037] **FIG. 19** illustrates a start menu in a full-screen mode in accordance with another embodiment of the present disclosure.

[0038] **FIG. 20** is block diagram illustrating a file manager that may be employed in a mobile device in accordance with some embodiments of the present disclosure.

[0039] **FIG. 21** is a block diagram illustrating an example of a user interface for providing a notification center on a mobile device in accordance with some embodiments of the present disclosure.

[0040] **FIG. 22** illustrates an example of a user interface for providing a security center on a mobile device in accordance with some embodiments of the present disclosure.

[0041] **FIG. 23** illustrates an example of a user interface for implementing a file manager in a mobile device in accordance with some embodiments of the present disclosure.

[0042] **FIG. 24** illustrates an example of a user interface for implementing page-switching on a mobile device in accordance with some embodiments of the present disclosure.

[0043] **FIG. 25** illustrates an example of mechanisms for user interfaces swap according to some embodiments of the present disclosure.

[0044] **FIG. 26** illustrates an example of a user interface for implementing a file manager in a mobile device in accordance with some embodiments of the present disclosure.

DETAILED DESCRIPTION

[0045] In the following detailed description, numerous specific details are set forth by way of examples in order to provide a thorough understanding of the relevant disclosure. However, it should be apparent to those skilled in the art that the present disclosure may be practiced without such details. In other instances, well known methods, procedures, systems, components, and/or circuitry have been described at a relatively high-level, without detail, in order to avoid unnecessarily obscuring aspects of the present disclosure. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present disclosure. Thus, the present disclosure is not limited to the embodiments shown, but to be accorded the widest scope consistent with the claims.

[0046] It will be understood that the term "system," "unit," "module," and/or "block" used herein are one method to distinguish different components, elements, parts, section or assembly of different level in ascending order. However, the terms may be displaced by other expression if they may achieve the same purpose.

[0047] It will be understood that when a unit, module or block is referred to as being "on," "connected to" or "coupled to" another unit, module, or block, it may be directly on, connected or coupled to the other unit, module, or block, or intervening unit, module, or block may be present, unless the context clearly indicates otherwise. As used herein, the term "and/or" includes any and all combinations of one or more of the associated listed items.

[0048] The terminology used herein is for the purposes of describing particular examples and embodiments only, and is not intended to be limiting. As used herein, the singular forms "a," "an," and "the" may be intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "include," and/or "comprising," when used in this disclosure, specify the presence of integers, devices, behaviors, stated features, steps, elements, operations, and/or components, but do not exclude the presence or addition of one or more other integers, devices, behaviors, features, steps, elements, operations, components, and/or groups thereof.

[0049] FIG. 1 is a block diagram illustrating an example 100 of a mobile device in accordance with some embodiments of the present disclosure. As shown, a mobile device 100 can include a display 110, a processor 120, an interface 130, a memory 140, and/or any other suitable component. In some embodiments, the mobile device 100 may be and/or include a mobile phone, a tablet computer, a wearable computing device, a Personal Digital Assistant (PDA), a Global Positioning System (GPS) device, a Personal Media Player (PMP), or any other apparatuses that have a display function. In some embodiments, the

mobile device 100 can include a touch screen to allow a user to input commands for executing programs with a pen, a finger, a stylus, or the like.

[0050] In some embodiments, the mobile device 100 can provide a multi-window display mode that allows multiple application windows to be displayed on the mobile device simultaneously.

[0051] In some embodiments, apart from the touch screen mentioned above, traditional input devices such as a keyboard, a mouse, or the like, may be employed to control the mobile device 100.

[0052] The display 110 can be any suitable device that is capable of receiving, converting, processing, and/or displaying media content and/or performing any other suitable functions. For example, the display 110 can be and/or include a Liquid Crystal Display (LCD) panel, Organic Light Emitting Diodes (OLED), a cathode ray tube (CRT) display, a plasma display, a touch-screen display, a simulated touch screen, the like, or any combination thereof. In some embodiments, the display 110 can be three-dimensional capable. In some embodiments, the display 110 can be implemented as a touch screen having inter-layer structure with a touch pad. For example, the display 110 can be used as the interface 130 as well as an output device, which will be described in more detail below. A touch screen can be configured to detect a touch input pressure, a touch input position, a touch input area, the like, or any combination thereof.

[0053] The display 110 can present media content (e.g., still images, video content, audio content, graphics, text, etc.) and/or any other information related to one or more applications running on the mobile device 100. For example, as will be discussed in more detail below, the display 110 may display user interfaces that provide a desktop, a start

menu, a task bar, a notification center, etc. on the screen. As another example, the display 110 can present one or more user interfaces (e.g., graphical user interfaces (GUIs)) provided by one or more applications running on the mobile device 100. Such a user interface may include any suitable media content. In some embodiments, user interfaces of multiple applications can be presented in multiple application windows. Each of the application windows may correspond to one of the applications. In a more particular example, as will be discussed in more detail in connection with FIG. 11 below, application windows 1110, 1120, and 1130 may be presented on a screen of a mobile device. In some embodiments, application windows 1110, 1120, and 1130 may correspond to a first application, a second application, and a third application running on the mobile device, respectively.

[0054] The application windows may be arranged on a screen of the mobile device 100 in any suitable manner. For example, each of the application windows may be implemented in any of a variety of sizes. In a more particular example, each of the application windows may or may not encompass the entire screen of the mobile device 100. In some embodiments, each of the application windows may be presented in one or more portions of the screen of the mobile device 100. In another more particular example, each of the application windows may have a particular size, such as a size that is defined by one or more coordinates and/or any other parameter that can be used to define the size of the application window. The coordinates and/or the parameter(s) may be provided by an operating system of the mobile device, the application, a user of the mobile device, etc. In some embodiments, the size of an application window may be dynamically adjusted based on one or more user inputs, such as a user input indicative of a user request to

maximize, minimize, dismiss, and/or restore the application window, a user input indicative of a user request to zoom the application window, etc. In some embodiments, multiple application windows may be selected, moved, etc. responsive to one or more user inputs. As will be discussed in more detail below, the user inputs may include and/or correspond to one or more gestures of a user of the mobile device 110, such as "touch," "press," "swipe," "drag," "double touch," "pinch," and/or any other gesture recognizable by the mobile device 100.

[0055] As another example, each of the application windows may be presented in any suitable position within the screen of the mobile device 100. In a more particular example, a given application window may or may not overlap with one or more portions of one or more other application windows. In another more particular example, an application window corresponding to an application can be presented in a portion of the screen defined based on positional information associated with the application and/or the application window (e.g., one or more coordinates provided by an operating system of the mobile device, a user, the application, etc.), one or more user inputs (e.g., a user input indicative of a request to move the application window, to adjust the size of the application window, etc.), and/or any other suitable information.

[0056] In some embodiments, one or more application windows may be arranged and/or be presented on a screen of mobile device 100 as described in connection with FIGS. 11-15 below.

[0057] The interface 130 can receive various user inputs. One or more of the user inputs may include and/or correspond to one or more gestures of a user of the mobile device

110, such as “touch,” “press,” “swipe,” “drag,” “double touch,” “pinch,” and/or any other gesture recognizable by the mobile device 100.

[0058] In some embodiments, the interface 130 can receive a user command to display the title area on application window when application window is displayed on maximization mode to encompass the whole screen area. The 'user command' can be user manipulation corresponding to touching a predetermined area of application window, such as a one-tap manipulation, and the like. The user command can be a flick manipulation, a drag manipulation, etc. In some embodiments, a certain area can be preset to be an information display area or a menu display area provided on, for example, the top of the application window, although not limited thereto.

[0059] When the maximized application window is displayed, the title area, displayed according to the user command, can provide a restoration button which can be used to restore the application window to mini mode or a dismissal button which can be used to close the application window. For example, in addition to the touch manipulation directed to the area excluding the area having buttons on the title area, a user command to restore maximization mode to mini mode can be received by a touch manipulation directed to the restoration button on the title area.

[0060] In some embodiments, the interface 130 can receive a user command to restore the mini mode by reducing the application window to a predetermined size while the the title area is displayed on the application window. The user command can be touch manipulation directed to the area excluding the area having buttons on the title area, or a touch manipulation directed to the window restore button provided on the title area.

[0061] The interface 130 can receive a user command to restore window to mini mode while the application window is in maximized state and while the title area is not displayed. The user command can be a two-tap manipulation, flick-down manipulation, drag-down manipulation directed to the information display area, etc.

[0062] The interface 130 can receive a user command to move application window position, a user command to adjust application window size, or a user command to pin-up application window on multi-window mode. The user command to pin-up can be implemented as a pin-up on/off by a tap manipulation directed to the pin-up button. Thus, when pin-up is on, the application window can be displayed in a fixed form. If pin-up is off, the application window can be implemented in a form in which moving position and adjusting size are possible.

[0063] The processor 120 can be and/or include any suitable hardware processor, such as a microprocessor, a micro-controller, digital signal processor, dedicated logic, and/or any other suitable circuitry for controlling one or more operations of one or more components of the mobile device 100, performing one or more operations described below in connection with FIGS. 9A-22, and/or performing any other function.

[0064] The processor 120 may control the display 110 to arrange and/or display one or more application windows on a screen based on a user command inputted through the interface 130.

[0065] Multiple application windows may be arranged within template screen of the mobile device according to an application executing order (e.g., the order in which application corresponding to the application windows are executed, activated, etc.), a frequency in which the application windows are displayed, time intervals during which the

application windows have been displayed, the number of the application windows, and the like. For example, a plurality of currently-operating application windows may be arranged and displayed according to a predetermined arrangement order that gives a higher priority to an application window with higher operation frequency. In some embodiments, the application windows may be arranged by performing one or more operations as described below in connection with FIGS. 9A-16.

[0066] If a predetermined user command is inputted, the processor 120 may provide a window edit mode screen so as to arrange a plurality of executed application windows according to a user selection on a window within the selected template. The predetermined user command may be a pinch manipulation on a screen. Specifically, the predetermined user command may be a pinch-in manipulation that uses multi-touch. Further, the window edit mode screen may close according to a pinch-out manipulation. The 'pinch-in manipulation' as used herein refers to manipulation of two fingers joining closer while 'pinch-out manipulation' as used herein refers to manipulation of the two fingers moving apart from each other. In some embodiments, another user manipulation such as flick-up and flick-down may be used, or another menu time to enter and exit window edit mode screen may be provided.

[0067] Further, on the window edit mode, currently executed applications may be displayed on the lower portion. Thus, a user may select an application and arrange the application on a desired area. In this case, an image corresponding to the application may be displayed on part of the corresponding application window area so that a user can recognize the application.

[0068] The processor 120 may control the mobile device 100 so that the application window is reduced to a predetermined size. In this case, a title area may be displayed in a fixed manner on one part of mini window area. Therefore, on the maximization mode in which the application window is displayed on the whole screen area, the title area may not be displayed basically and may be displayed according to a user command. However, the title area may be displayed basically in a fixed manner. According to an exemplary embodiment of the present invention, the title area may be displayed according to a user command even on the mini mode.

[0069] Further, the processor 120 may control the mobile device 100 so that a guide GUI is displayed to provide guide about position movement and size adjustment of the application windows on the multi-window mode according to a user command. In this case, the processor 120 may control the guide GUI to provide haptic feedback on an area on which the guide GUI is displayed.

[0070] The memory 140 may be and/or include any hardware device that can store information for use in the mobile device 100, such as random-access memory (RAM), read-only memory (ROM), programmable read-only memory (PROM), EEPROM, flash memory, hard disk drives, solid-state drives, etc. The memory 140 may store information about applications, application windows, and/or any other information to implement a mobile operating system. For example, the memory 140 may store one or more configuration lists and or configuration files that may be used to present application windows. memory 140.

[0071] It should be noted that the description of the mobile device above is provided for the purposes of illustration, and not intended to limit the scope of the present disclosure.

For persons having ordinary skills in the art, a certain amount of variations and modifications may be conceptualized and reduced to practice in the light of the present disclosure. However, those variations and modifications may not depart from the protecting scope of the present disclosure.

[0072] FIG. 2 is a flowchart illustrating an input execution by the mobile device 100 in accordance with some embodiments of the present disclosure. In step 201, input may be received by the mobile device 100. The input can be 'user command' that can be user manipulation corresponding to touching a predetermined area of application windows, such as a one-tap manipulation, and the like. Alternatively, the user command can be a flick manipulation or a drag and drop manipulation. In some embodiments, the user command can be a mouse click, keyboard input, gesture input, a voice control command, and the like. The voice control command may be configured to open an application, manipulate an application window, and the like.

[0073] In some embodiments, user inputs received by a mouse, a keyword, a microphone, and/or any other input device may be converted into one or more gestures recognizable by the mobile device. For example, a user input (e.g., "click") captured by a mouse may be converted in a gesture (e.g., "touch"). The conversion may be performed by mapping the user input to the gesture based on information about predetermined mapping information. Similarly, a user gesture may be mapped to a mouse input or any other type of user input based on the mapping information.

[0074] The input received in step 201 may be processed in step 202. The processing can recognize the input perform based on the input.

[0075] The memory can be accessed in step 203. In some embodiments, an application configuration list can be stored in the memory 140. The size of application window for an application may be stored in the application configuration list. When an icon corresponding to an application is touched or clicked, the touch or the click can be processed and recognized. The application configuration list may be queried and the size of the application window for the application can be extracted.

[0076] An output can be generated in step 204. In some embodiments, when an application is clicked or touched, a container aimed to contain the application and an application window for displaying the application can be generated.

[0077] The output generated in step 204 can be displayed in step 205. In some embodiments, an application window for an application can be displayed in a display.

[0078] Once step 205 is completed, it can return to step 201 and continuously receive input.

[0079] It should be noted that the flowchart described above is merely provided for the purposed of illustration, and not intended to limit the scope of the present disclosure. For persons having ordinary skills in the art, various variations and modifications may be conducted under the teaching of the present disclosure. However, those variations and teachings may not depart from the scope of the present disclosure.

[0080] FIG. 3 is a block diagram of the mobile device 100 in accordance with some embodiments of the present disclosure. As illustrated, mobile device 100 may include a display 110, a processor 120, a user interface 130, a memory 140, an audio processor 310, a video processor 320, a power supply unit 330, a sensor module 340, a I/O module 350, a touch screen controller 360, a GPS module 370, an application driver 371, a feedback

provider 372, a mobile communication module 380, and a camera module 390. With the components described in FIG. 3, the description of the components with reference to FIG. 1 will not be further explained as such an explanation would be unnecessary.

[0081] The operation of the processor 120 may be performed by programs stored in the memory 140. The memory 140 may store various data such as an Operating System (OS) software module to drive the mobile device 100, various applications, various data inputted or established while executing applications, or content. Particularly, depending on the characteristics thereof, the applications stored in the memory 140 may be classified into applications that support multi-window mode and other applications that do not support the multi-window mode.

[0082] Further, the memory 140 may store information related to layout to arrange one or more application windows. In some embodiments, the memory may store an application configuration list and/or application type list as described in connection with FIG. 8 below.

[0083] Other various software modules stored in the memory 140 will be described below by referring to FIG. 3.

[0084] The GPS module 370 may receive radio waves from a plurality of GPS satellites (not shown) in an Earth orbit, and calculate a location of the mobile device 100 by using a time of arrival of the radio waves from the GPS satellites (not shown) to the mobile device 100.

[0085] The application driver 371 may drive and implement applications that the mobile device 100 can provide. The 'application' as used herein may refer to the programs that can automatically run which may include various multimedia content. The term 'multimedia content' as used herein includes text, audio, still images, animation, video,

interactivity content, Electronic Program Guide (EPG) content from content provider, electronic message received from users, or information regarding current events, or the like.

[0086] The feedback provider 372 may perform a function of providing various feedbacks according to functions implemented in the mobile device 100.

[0087] For example, the feedback provider 372 may provide haptic feedback regarding a GUI displayed on a screen. The haptic feedback technology, or computer touch technology enables users to feel touch, by generating vibration, power or impact in the mobile device 100. The feedback provider 372 may provide haptic feedback regarding guide a GUI, if the guide GUI is displayed to provide guide about position movement and size adjustment of application windows according to user command on the multi-window mode.

[0088] Under this circumstance, the feedback provider 372 may provide various feedbacks by differently applying vibrating conditions such as a vibrating frequency, a vibrating length, a vibrating shape, a vibrating position, and the like according to controlling of the processor 120. A method of generating various haptic feedback by differently applying vibration corresponds to technology according to the related art, and will not be further explained for the sake of brevity.

[0089] Although the feedback provider 372 may use a vibration sensor and provide haptic feedback, in some embodiments haptic feedback may be provided by using a piezo sensor.

[0090] The mobile communication module 380 may be configured to communicate with various types of external apparatuses according to various types of communication

methods. The mobile communication module 380 may include various communication chips such as a Wi-Fi chip 381, a Bluetooth™ chip 382, a Near Field Communication (NFC) chip 383, a wireless communication chip 384, and the like.

[0091] The Wi-Fi chip 381 and the Bluetooth™ chip 382 may communicate according to Wi-Fi method and Bluetooth™ method, respectively. The wireless communication chip 384 may communicate according to various communication protocols such as Institute of Electrical and Electronics Engineers (IEEE) standards, ZigBee, 3rd Generation (3G), 3rd Generation Partnership Project (3GPP), or Long Term Evolution (LTE). The NFC chip may operate according to NFC method using 13.56 MHz from the various RF-ID frequency bandwidths such as 135 kHz, 13.56 MHz, 433 MHz, 860-960 MHz, and 2.45 GHz.

[0092] The audio processor 310 may be configured to process audio data. In the audio processor 310, various audio data processing such as decoding, amplifying, or noise filtering may be implemented.

[0093] The video processor 320 may be configured to process video data. In the video processor 320, various video data processing such as decoding, scaling, noise filtering, converting frame rate, or converting resolution may be implemented.

[0094] The power supply unit 330 may supply power to a single battery or a plurality of batteries (not shown) disposed in the housing of the mobile device 100 according to the control of the processor 120. The single battery supplies or the plurality of batteries (not shown) supply power to the mobile device 100. The power supply unit 330 may supply power input from an external power source (not shown) to the mobile device 100 over the wired cable connected to the connector 355.

[0095] The sensor module 170 includes at least one sensor that detects a state of the device 100. For example, the sensor module 170 may include a proximity sensor that detects whether the user approaches the device 100, an illumination sensor (not shown) that detects an amount of light in the vicinity of the device 100, or a motion sensor (not shown) that detects an operation (for example, a rotation of the device 100, an acceleration or vibration applied to the device 100) of the device 100. The at least one sensor may detect the state of the device 100, generate a signal corresponding to the detected state, and transmit the signal to the control unit 110. The sensor of the sensor module 170 may be added or deleted according to the performance of the device 100.

[0096] The I/O module 350 may be configured to input or output information. The I/O module 350 may include a speaker 351, buttons 352, a USB port 353, a microphone 354, a connector 355, a vibration motor 356, a key pad 357, and the like.

[0097] The speaker 351 may be configured to output various alarm sounds and voice messages as well as various audio data processed in the audio processor 310.

[0098] The buttons 352 may correspond to various types of buttons such as a mechanical button, a touch pad, or a wheel installed on a random area of front face, side face, or back face of a body of the mobile device 100. For example, a button to turn electrical power on/off may be installed on the mobile device 100.

[0099] The USB port 353 may perform communication with various external apparatuses through USB cables or charge electric power.

[00100] The camera module 390 may include at least one of the first and second cameras 391 and 392 that capture a still image or a moving image according to the control of the processor 120. The first camera 391 or the second camera 392 may include an auxiliary

light source (for example, a flash (not shown)) that provides an amount of light used for photographing. The first camera 391 may be disposed in the front of the mobile device 100. The second camera 392 may be disposed in the rear of the mobile device 100. In some embodiments, the first camera 391 and the second camera 392 may be disposed to be adjacent to each other (for example, an interval between the first camera 391 and the second camera 392 may be from 1 cm to 8 cm) and may capture a 3 Dimensional (3D) still image or a 3D moving image.

[00101] The microphone 354 may be configured to receive user voices or other sounds and to convert the received sound into audio data. The processor 120 may use user voices inputted through the microphone 354 during a call. The processor 120 may convert user voices into audio data and store the audio data in the memory 140.

[00102] The connector 355 may be used as an interface for connecting the device 100 and the external apparatus (not shown) or a power source (not shown). Data stored in the storage unit 175 of the device 100 may be transmitted to the external apparatus (not shown) or data may be received from the external apparatus (not shown) over a wired cable connected to the connector 355 according to the control of the control unit 110. Power is input from the power source (not shown) or a battery (not shown) may be charged over wired cable connected to the connector 355.

[00103] The vibration motor 356 may convert the electrical signal into a mechanical vibration according to the control of the control unit 110. For example, in a case where the device 100 in a vibration mode receives the voice call from another apparatus (not shown), the vibration motor 356 operates. A single vibration motor 356 or a plurality of vibration motors 164 may be formed in the housing of the device 100. The vibration motor 356 may

operate in response to a user touch operation that touches the touch screen 190 and a continuous touch motion on the touch screen 190.

[00104] The key pad 357 may receive a key input from a user to control the device 100. The key pad 357 includes a physical key pad (not shown) formed in the device 100 or a virtual key pad (not shown) displayed on the touch screen 190. The physical key pad (not shown) formed in the device 100 may be excluded according to a performance or structure of the device 100.

[00105] If the camera module 390 and the microphone 354 are provided, the processor 120 may perform controlling operation according to user voices inputted through the microphone 354 or according to user motions recognized by the camera module 390. For example, the mobile device 100 may operate on motion controlling mode and voice controlling mode. If operating on motion controlling mode, the processor 120 may activate the camera module 354 to photograph a user, trace changes in user motions and perform corresponding controlling operations. If operating on voice controlling mode, the processor 120 may analyze user voices inputted through the microphone and perform controlling operations according to the analyzed user voices (e.g., may operate on voice recognizing mode).

[00106] In some embodiments, the mobile device 100 may include various external inputting ports to connect various external components such as a headset, a mouse, a LAN, and the like.

[00107] The processor 120 may control the overall operation of the mobile device 100 by using various programs stored in the memory 140.

[00108] For example, the processor 120 may implement applications stored in the memory 140, create an implementing screen, and display the screen. Further, the processor 120 may play various content stored in the memory 140. The processor 120 may also communicate with external apparatuses through the mobile communication module 380.

[00109] In some embodiments, the processor 120 may include a Random Access Memory (RAM) 121, a Read Only Memory (ROM) 122, a Central Processing Unit (CPU) 123, a graphics processing unit (GPU) 124, first to (n)th interfaces 125-1~125-n, a bus 126, and the like.

[00110] The RAM 121, the ROM 122, the CPU 123, the GPU 124, and the first to (n)th interfaces 125-1~125-n may connect to each other through the bus 126.

[00111] The first to (n)th interfaces 125-1~125-n may connect to the sorts of components as described above. In some embodiments, one of the interfaces may be a network interface connecting to external apparatuses through network.

[00112] The CPU 123 may access the memory 140, and perform booting by using the Operating System (OS) stored in the memory 140. The CPU 123 may perform various operations by using various programs, content, and data stored in the memory 140.

[00113] The ROM 122 may store therein a set of commands to boot the system. If a command to turn-on is inputted and if electrical power is provided, the CPU 123 may load the OS stored in the memory 140 to the RAM 121according to commands stored in the ROM 122, run the OS, and boot the system. If booting completes, the CPU 123 may paste various application programs stored in the memory 140 to the RAM 121, implement the application programs pasted on the RAM 121, and perform various operations.

[00114] The GPU 124 may create a screen including various objects such as icons, images, or text by using a calculator (not illustrated) and a renderer (not illustrated). The calculator may calculate attribute values such as coordinates, shapes, sizes, or colors of each object to be displayed according to screen layout by using a control command received from the I/O module 350. The renderer may create a screen of various layouts including objects based on the attribute values calculated by the calculator. The screen created by the renderer may be displayed within a display area of the display 110.

[00115] The touch screen controller 360 may convert the analog signal received from a touch screen (not shown in the figure) into a digital signal (for example, X and Y coordinates) and transmits the digital signal to the processor 120. The processor 120 may control the touch screen by using the digital signal received from the touch screen controller 360. For example, the processor 120 may select or execute a shortcut icon (not shown) displayed on the touch screen in response to the touch. In some embodiments, the touch screen controller 360 may be included in the processor 120.

[00116] In some embodiments, the mobile device 100 may additionally include a detector (not illustrated). The detector (not illustrated) may detect various manipulations of the mobile device 100 such as touching, rotating, sliding, pressing, and zooming.

[00117] In some embodiments, the detector (not illustrated) may include a touch sensor for detecting touch. A touch sensor may be implemented as a capacitive type or a resistive type. The capacitive type sensor uses coated conductive material on the surface of the display 150, detects micro electricity excited by movement of a user when a user touches the surface of the display 110 (e.g., with the user's hand, a pen, or the like), and calculates touch coordinate. The resistive type sensor includes two electrode panels, detects the

electrical current flow caused by contacting upper panel and lower panel on a touch point if a user touches the screen, and calculates touch coordinate. The touch sensor may be implemented with various types as explained above. The detector may further include a geomagnetic sensor to detect rotating situation and moving direction of the mobile device 100, an acceleration sensor to detect sliding degree of the mobile device 100, and the like.

[00118] It should be noted that the description above is provided merely for the purposes of illustration, and not intended to limit the scope of the present disclosure. For persons having ordinary skills in the art, various variations and modifications may be conducted under the teaching of disclosure provided herein. However, those teachings may not depart from the protecting scope of the present disclosure. For example, a Global Positioning System (GPS) receiver (not shown in the figure) for receiving GPS signals from the GPS satellite and for calculating current position of the mobile device 100, or a Digital Multimedia Broadcasting (DMB) receiver (not shown in the figure) for receiving and processing DMB signals may be further included.

[00119] FIG. 4 is a block diagram illustrating an example 400 of a mobile operating system (OS) that may be employed in the mobile device 100 in accordance with some embodiments of the present disclosure. In some embodiments, the mobile OS 400 may include Android™ or any other mobile OS.

[00120] As illustrated in FIG. 4, OS 400 includes a set of C/C++ libraries in libraries layer 430 that are accessed through application framework layer 420. Libraries layer 430 includes the "bionic" system C library 436 that was developed specifically for Android™ to be smaller and faster than the "glibc" Linux C-library. Libraries layer 430 also includes inter-process communication ("IPC") library 433, which includes the base classes for the

"Binder" IPC mechanism of the Android™ OS. Binder was developed specifically for Android™ to allow communication between processes and services. Other libraries shown in Libraries layer 430 in FIG. 4 include media libraries 432 that support recording and playback of media formats, surface manager 431 that managers access to the display subsystem and composites graphic layers from multiple applications. Other libraries that may be included in Libraries layer 430 but are not pictured in FIG. 4 include bitmap and vector font rendering libraries, utilities libraries, browser tools (e.g., WebKit, etc.), and/or secure communication libraries (e.g., SSL, etc.).

[00121] Application framework layer 420 of OS 400 provides a development platform that allows developers to use components of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, etc. Application framework layer 420 also allows applications to publish their capabilities and make use of the published capabilities of other applications. Components of Application framework layer 420 of OS 400 include activity manager 421, resource manager 422, window manager 423, dock manager 424, hardware and system services 425, desktop monitor service 426, location manager 427, and remote communication service 428. Other components that may be included in application framework layer 420 of OS 400 and not shown in the figure may include a view system, telephony manager, package manager, location manager, and/or notification manager, among other managers and services.

[00122] Applications running on OS 400 run within the virtual machine (VM) 441 in the runtime environment 440 on top of the object-oriented application framework. VM 441 may be a register-based virtual machine, and runs a compact executable format that is designed to reduce memory usage and processing requirements. Virtual machine 441 may

be and/or include the Dalvik virtual machine in some embodiments. Applications running on OS 400 include desktop 411, start menu 412, file manager 413, browser 414, and/or other application(s) ("App(s)") 415.

[00123] As referred to herein, an activity represents a single screen with a user interface. An application, or an application window may include one or more activities. For example, a message application may have one activity showing a list of contacts, another activity to draft a new message, and another activity for reading received messages. It should be noted that each activity is independent and an activity of an application may be activated by another activity in another application, for example, a drafting new message activity may be activated by a camera application as a user may choose to share his newly taken photos.

[00124] The mobile OS may monitor all the activities by considering the lifecycles of the activities and/or their corresponding application. For example, the mobile OS can monitor the time an activity can remain invisible or inactive before being shut down by the system.

[00125] In some embodiments, the mobile OS may set an activity status for each of the running activities. The activity status may be and/or include an "onCreate()" status, an "onStart()" status, an "onResume()" status, an "onPause()" status, an "onStop()" status, an "onDestroy()" status, etc. Generally, the "onCreate()" status may be assigned to an activity when it's just created. The status may be changed to an "onStart()" status when there is enough system memory for running the activity. If the activity is in the foreground of the screen (at the top of the stack/layer), it is active or running. However, if a new non-full-sized or transparent activity has been focused, e.g., been lately interacted by users, on top of the activity, while the activity is at least partially visible, an Onpause() status may be

assigned to the activity. This status may indicate that the activity may maintain all state and member information and remain attached to the window manager, but may be killed by the system in extreme low memory situations. If the activity is completely obscured by another activity, it may be terminated. The Activity may still retain all state and member information, however, it may no longer be visible to the user so its window is hidden and it may often be killed by the system when memory is needed elsewhere. If an activity is paused or stopped, the system can drop the activity from memory by either asking it to finish, or simply killing its process. When it is visible again to the user, e.g., when all the activities above it are closed or shut down, or when it is either resumed or restarted, it must be completely restarted and restored to its previous state before displaying to the user.

[00126] In some embodiments, if an activity is opened and placed to the foreground that it partially overlap on a previous activity, the previous activity is ought to be placed in an Onpause() status because it still appears partially visible to the user. However, because it is not actually a "window", or in other word, because the system believes that the previous activity is being displayed in a full-screen mode and is being completely obscured by the new activity, an Onstop() status may be assigned to the activity and the activity may be shut down by the system when memory is needed elsewhere. This may be a huge problem because one of the benefits of running activities in windows mode under Android™ OS is to enable multi-task and multi-window processing, and putting the old activities in Onstop() status may just prohibit running more than two activities and lose the system this benefit as the old activities may be directly shut down as soon as new activities are opened.

[00127] FIG. 5 illustrates an example 500 of a kernel according to various embodiments. Kernel 500 may include touch-screen display driver 510, camera driver(s) 520,

Bluetooth™ driver(s) 530, shared memory allocator 540, IPC driver(s) 550, USB driver(s) 560, WiFi driver(s) 570, I/O device driver(s) 580, power management module 590, and/or any other suitable component for implementing various functions of an operating system. I/O device driver(s) 580 may include device drivers for external I/O devices, including devices that may be communicatively coupled to the mobile device 100 through an interface among first interface 125-1 to (n)th interface 125-n. Kernel 500 may include other drivers and functional blocks including a low memory killer, kernel debugger, logging capability, and/or other hardware device drivers.

[00128] FIG. 6 illustrates a mobile OS that can be employed in the mobile device 100 and its interactions with kernel and hardware in accordance with some embodiments of the present disclosure.

[00129] As illustrated in FIG. 6, mobile OS 600 has libraries layer 630, application framework layer 620, and application layer 610. In mobile OS 600, applications 611 and 612 run in application layer 610 supported by application framework layer 620 of mobile OS 600. Application framework layer 620 may include manager(s) 621 and service(s) 622 that are used by applications running on mobile OS 600. For example, application framework layer 620 may include window manager, activity manager, package manager, resource manager, telephony manager, gesture controller, and/or other managers and services for the mobile environment. Application framework layer 620 may include a mobile application runtime environment that executes applications developed for mobile OS 600. The mobile application runtime environment may be optimized for mobile computing resources such as lower processing power or limited memory space. The mobile application runtime environment may rely on the kernel for process isolation,

memory management, and threading support. Libraries layer 630 includes user libraries 631 that implement common functions such as I/O and string manipulation ("standard C libraries"), graphics libraries, database libraries, communication libraries, and/or other libraries.

[00130] FIG. 7 illustrates a multi-window framework architecture that can be employed by the mobile device 100 in accordance with some embodiments of the present disclosure. The multi-window framework may be configured to display a plurality of application windows on a screen, for example, the display 110 or the user interface 130.

[00131] As illustrated, the multi-window framework architecture may include an application framework 730 and an application layer 710. In this case, the multi-window framework 720 may operate inside the application framework 730. In some embodiments, the multi-window framework 720 may operate separately from the application framework. It should be noted that the multi-window framework described herein is compatible with the mobile OS described in FIGS. 4, 5, and 6.

[00132] The application framework 730 may include a multi-window framework 720, an activity manager 731, a window manager 732, a content provider 733, and a view system 734, a package manager 735, a telephony manager 736, a resource manager 737, a location manager 738, a notification manager 739, and the like. The multi-window framework 720 may include a multi-window manager 721 and a multi-window service 722.

[00133] As applications are executed, the activity manager 731 may call information corresponding to an application window of the executed application to the multi-window framework. The activity manager 731 may receive information regarding display mode, size and coordinate of an application window based on a life cycle of the application

window from the multi-window framework. The activity manager 731 may call information regarding display mode, size and coordinate of the application window during creating step of the life cycle of the application window.

[00134] Further, the window manager 732 may confirm the application window corresponding to a touch inputted by a user. The window manager 732 may provide position information on a display corresponding to a touch input of a user to the multi-window framework, and receive information of the application window corresponding to the touch input determined by the multi-window framework from the multi-window framework.

[00135] According to exemplary embodiments of the present invention, in response to a touch input of a user, the window manager 732 may receive information regarding coordinate and size of application window from the multi-window framework, and determine application window corresponding to touch input of a user based on the received position and size of application window.

[00136] The view system 734 may confirm coordinates and sizes of widget window and pop-up window. In this case, the multi-window framework 720 may determine sizes and coordinates of widget window and pop-up window, and the view system 734 may receive information regarding sizes and coordinates of widget window and pop-up window from the multi-window framework 720.

[00137] The multi-window manager 721 included in the multi-window framework 720 can manage various operations regarding multi-window functions provided from the mobile device 100, and can provide various Application Programming Interfaces (APIs) regarding multi-window functions. Further, the multi-window service 722 may store

various APIs regarding multi-window functions. An API regarding common functions of single window and multi-window may be embodied as common class, and an API regarding functions only applied in multi-window may be implemented so as to be divided according to display mode.

[00138] In some embodiments, the multi-window framework 720 may communicate with the window manager 732, the view system 734, and/or the activity manager 731 to control presentation of one or more application windows. For example, the activity manager 731, the view system 734, the window manager 732, and the multi-window manager 721 may be configured to call APIs for performing one or more functions to cause multiple application windows corresponding to applications on the screen of the mobile device. The multi-window manager 721 may provide a function of the multi-window service 722 via an API. The multi-window service 722 can track the lifecycle of one or more of the applications and can manage display parameters of the applications. For example, the called API may be used to manage the size (e.g., a size defined by a width, a height, etc.), positional information (e.g., one or more coordinates), visibility to a user, and/or any other information of an application window. In some embodiments, the application layer 710 may call APIs from the multi-window framework directly.

[00139] The operating system of the mobile device may assign a default size of application window to the application. The default size may be assigned when the application is installed, activated, etc. on the mobile device. The default size may be represented by a default width, a default height, and/or any other suitable parameters.

[00140] In some embodiments, the default size may be assigned based on the type of the application (e.g., a phone application, a tablet application, etc.). The type of the

application may indicate one or more types of mobile devices and/or platforms on which the application is designed to run. For example, the type of an application may indicate that the application is a phone application that is designed to run on a mobile phone. As another example, the type of an application may indicate that the application is a tablet application that is designed to run on a tablet computer. In some embodiments, a first default width and a first height may be assigned to phone applications running on a particular mobile device (e.g., a tablet computer, a mobile phone, etc.). The first default width and the first height may form an aspect ratio that is suitable for presentation of application windows of the phone applications based on the display resolution of the mobile device. In some embodiments, a second default width and a second height may be assigned to tablet application running on the particular mobile device. The second default width and the second height may form an aspect ratio that is suitable for presentation of application windows of the tablet application based on the display resolution of the mobile device.

[00141] As will be described in more detail in connection with FIG. 8, application windows of phone applications may be presented on a tablet computer in a suitable manner based on the default size assigned to the phone applications. Similarly, application windows of tablet applications may be presented on a mobile phone in a suitable manner based on the default size assigned to the tablet applications. As such, application windows and user interfaces of various types of applications may be presented on mobile devices with different display parameters in a proper manner.

[00142] In some embodiments, upon presenting an application window of an application on a mobile device, the multi-window manager 721 may monitor information about the

size, position, etc. of the application window and may store such information. For example, the multi-window manager 721 may monitor and store positional information (e.g., one or more coordinates), one or more widths and/or heights of the application window, and/or any other information about the application window when a user of the mobile device interacts with the application window (e.g., by zooming, dragging, selecting, minimizing, dismissing, etc. the application window). In some embodiments, the multi-window manager 721 may monitor and store information about the position and the size of the application window in the memory 140 when the application window is dismissed and/or minimized. The application window may be presented based on the recorded position and/or size when the application window is later presented on the mobile device.

[00143] When an application is activated, the application may request one or more display parameters from the multi-window manager 721 (e.g., by making one or more API calls). The display parameters may include the display resolution of the mobile device, the pixel density of the mobile device, a scaling factor for fonts displayed on the mobile device, and/or any other information about the information about the display of the mobile device. Alternatively or additionally, the display parameters may also include one or more parameters related to the size, the position, and/or any other information about an application window which corresponds to the application.

[00144] In some embodiments, the request for display parameter(s) may include and/or be a request for a display resolution of the mobile device. Upon receiving the request, the multi-window manager 721 may determine a size of application window associated with the application. The size of application window may be, for example, a default size (e.g., a default width, a default height, etc.), a recorded size (e.g., the size in which an application

window of the application was previously presented on the mobile phone), and/or any other suitable size of application window associated the application. In some embodiments, upon receiving the request for the display resolution, the multi-window manager 721 may determine whether any recorded size is associated with the application. In response to determining that a recorded size is associated with the application, the multi-window manager 721 may return information about the recorded size (e.g., a recorded width, a recorded height, etc.) responsive to the request. Alternatively, in response to determining that no recorded size is associated with the application, the multi-window manager 721 may return information about the default size (e.g., the default width, the default height, etc.) responsive to the request.

[00145] In a more particular example, when an activity of an application is created and started, the operation system of a mobile device (e.g. an OS based on the Android™ OS) may first assign a default size for displaying the activity. In some embodiments, the activity may be displayed in the default size using one or more of the function getMetrics () the function DisplayMetrics (), etc. defined in the Android™ OS. The function getMetrics () may be used to return display parameters and/or metrics to the multi-window manager. The function DisplayMetrics() may be used to return the display resolution of the mobile device and to adjust a size in which an application window is presented accordingly.

[00146] When an application calls the DisplayMetrics() for the display resolution of the mobile device, the Android™ system would return default variables regarding the display resolution to the application. As such, any activity in the Android™ system is displayed

in a full-screen and it may eventually convert to full screen even if it is not originally opened in a full-screen at the beginning.

[00147] In some embodiments, in order to display the activity in an application window or the like that does not encompass the whole screen of the mobile device, a "windowlizing" process may be implemented to present windows of applications and/or activities in a window or similar visual representations.

[00148] For example, when an application or an activity is created and started, default sizes and/or positional information of application windows corresponding to the application and/or activity may be determined. Additionally, a size in which an application window was previously presented can be stored as a recorded size. When the application is to provide an application window for presentation on the mobile device (e.g., in a full-screen mode), the application may call the function DisplayMetrics () to request the display resolution of the mobile device. The multi-window manager 721 may return the recorded size to the application. The application window may then be presented in the recorded size. Alternatively, the multi-window manager 721 may return the default size to the application so that the application window may be presented in the default size. As such, the application window may be presented in the recorded size, the default size and/or any other size defined by the multi-window manager, while the application requests the application window be presented in a full-screen mode.

[00149] In some embodiments, upon receiving the request for the display resolution, the multi-window manager 721 may determine that an application window of the application is to be presented in a full-screen mode (e.g., by encompassing the entire screen of the mobile device). In some embodiments, this determination may be made upon receiving a

request to maximize the application window, a request to present media content (e.g., video content, images, etc.) in a full-screen mode, etc. In response to making such a determination, the multi-window manager 721 may return information about the display resolution of the mobile device responsive to the request for the display resolution.

[00150] In some embodiments, a desktop program, a start menu program, a task bar program may be included in the application layer 710. The desktop program, the start menu program, and the task bar program may communicate with the multi-window framework to provide one or more user interfaces, such as one or more desktop interfaces, start menu interfaces, a task bar, etc.

[00151] Further in some embodiments, a notification center program and a security center program may be included in the application layer 710. The notification center program and the security center program may communicate with the multi-window framework directly as well, for example, to provide various user interfaces for presentation on the mobile device.

[00152] It should be noted that the description of the multi-window framework is provided for the purposes of illustration, and not intended to limit the scope of the present disclosure. For persons having ordinary skills in the art, various modifications and variations may be conducted under the teaching of the present disclosure. However, those teachings may not depart from the protecting scope of the present disclosure.

[00153] FIG. 8 is a block diagram illustrating an application configuration list and an application type list that can be employed in the mobile device 100 in accordance with some embodiments of the present disclosure. The application configuration list 810 and the application type list 820 can be stored in any suitable storage device, such as a local

storage (e.g., the memory 140 of FIG. 1), a hard drive, a solid state storage device, a removable storage device a cloud-based storage device, and/or any other device that can store the configuration lists.

[00154] The application configuration list 810 can include data item 811, data item 812, etc. While two data items are illustrated in FIG. 8, this is merely illustrative. It should be noted that the application configuration list 810 can include any suitable number of data items. In some embodiments, the application configuration list 810 can include two or more data items (e.g., 10 items, 100 items, etc.). In some embodiments, the number of data items included in the application configuration list 810 may relate to the number of applications installed on the mobile device 100. Each data item in the application configuration list 810 may store information relating to the size of application windows of an application, positional information about application windows of an application, application settings, application usage record, and/or any other information related to the application.

[00155] For example, data items 811 and 812 can store information about the size of application windows for App 1 and App 2, respectively. More particularly, for example, data item 811 may store information about a default size of application window for App 1 (e.g., a default width, a default height, etc.), a size of the application window when the application was previously activated, a size defined by a user, etc. When an application (e.g., App 1) is activated, an application window of App 1 may be presented on the mobile device in accordance with the information stored in data item 811.

[00156] As another example, data items 811 and 812 can store positional information about application windows for App 1 and App 2, respectively. The positional information

about an application may include, for example, one or more coordinates based on which the application window may be displayed on a screen of the mobile device.

[00157] When an application is activated, the size and/or positional information of an application window of the application may be accessed and extracted to display the application window. In some embodiments, the size and the coordinate of the application window may be accessed and/or extracted by the multi-window manager 721 or the multi-window service 722 as described above.

[00158] The application type list 820 can include data item 821, data item 822, etc. The application type list 820 can include any suitable number of data items (e.g., 1 item, 2 items, 10 items, 100 items, etc.). In some embodiments, the number of items included in the application type list 820 may relate to the number of applications installed on the mobile device 100. Each of the data items in the application type list may correspond to one or more applications. The data items in the application type list 820 may store information relating to the type of one or more applications (e.g., a phone application, a tablet application, etc.). For example, data item 821 can store information about the type of App 1. As another example, data item 822 can store information about the type of App 2.

[00159] In some embodiments, the type of an application may indicate one or more types of mobile devices and/or platforms on which an application is designed to run. For example, the type of an application may indicate that the application is a phone application that is designed to run on a mobile phone. As another example, the type of an application may indicate that the application is a tablet application that is designed to run on a tablet computer.

[00160] In some embodiments, one or more application windows of an application may be presented on a mobile device based on information about the type of the application. For example, one or more application windows of a phone application may be presented on a tablet computer in a default manner that is suitable for the screen of the tablet computer. In a more particular example, instead of being displayed in a default full-screen mode suitable for a mobile phone, an application window of the phone application may be presented on the tablet computer in a particular size that is suitable for the screen of the tablet computer (e.g., a size defined by a predetermined width, a predetermined height, etc.). In another more particular example, an application window of the phone application may be presented on the screen of the tablet computer in a particular orientation that is suitable for the screen of the tablet computer (e.g., horizontal, reverse horizontal, vertical, reverse vertical, etc.). As such, a user may not need to rotate the tablet computer when interacting with the phone application using the tablet computer.

[00161] As another example, one or more application windows of a tablet application may be presented on a mobile phone in a default manner that is suitable for the screen of the mobile phone. In a more particular example, instead of being displayed in a default full-screen mode suitable for a tablet computer, an application window of the tablet application may be presented on the mobile phone in a particular size that is suitable for the screen of the mobile device (e.g., a size defined by a predetermined width, a predetermined height, etc.). In another more particular example, an application window of the tablet application may be presented on the screen of the mobile phone in a particular orientation that is suitable for the screen of the mobile phone (e.g., horizontal, reverse horizontal, vertical, reverse vertical, etc.).

[00162] It should be noted that the description of the application configuration list and the application type list above are provided for the purposes of illustration, and not intended to limit the scope of the present disclosure. For persons having ordinary skills in the art, a certain amount of variations and modifications may be conceptualized and reduced to practice in the light of the present disclosure. However, those variations and modifications may not depart from the protecting scope of the present disclosure. For example, the application configuration list and the application type list can be merged into a single list.

[00163] In some embodiments, when an application is activated, the size and coordinate of the application window may be adjusted based on information the application configuration list that corresponds to the application. Alternatively or additionally, the application window may be adjusted based on one or more default settings (e.g., when information about the application cannot be found on the list). The default settings may include a default size, one or more default coordinates, and/or any other default parameter of one or more application windows.

[00164] In some embodiments, when an application is closed after use, a configuration file may be used to store the most previously used size and positional information of the application window. In some embodiments, when an application window of the same application is presented later on the mobile device, the application may query the configuration file to extract the size and the coordinate for display. If the configuration file is either lost upon restart or erased by user, the application may use the application configuration list and may present the application based on the configuration list as described above.

[00165] In some embodiments, the application configuration list mentioned above may be stored in a cloud based server which updates regularly or in the system hardware, for example, the memory 140.

[00166] In some embodiments, the application configuration list may contain information relating to whether an application is a mobile or tablet based application, and the size of the application window for the application may be set accordingly. In some embodiments, the application configuration list may contain whether maximization, minimization, restoration, and/or any other function are enabled by the application. In still some embodiments, the application configuration list may contain the types of the applications that the size of applications may be adjusted according to their corresponding types.

[00167] In alternative embodiments, machine learning methods may be implanted into the application configuration list so that any application which is not on the application configuration list may be classified into a certain type in the list and be displayed according to the corresponding size of that type stored in the application configuration list.

[00168] FIG. 9A is a flow diagram illustrating an example 900 of a process for presenting multiple application windows on a mobile device in accordance with some embodiments of the present disclosure.

[00169] As illustrated, process 900 may begin by receiving a request to activate one or more applications on the mobile device at 905. The request may correspond to one or more user inputs that initiate an interaction with the application(s) (e.g., interacting content provided by the application(s), sharing content with other users using the application(s), a request to launch the application(s), etc.). In some embodiments, multiple requests to

activate multiple applications may be received sequentially, simultaneously, substantially simultaneously, or in any other suitable manner.

[00170] At 910, process 900 can present one or more application windows corresponding to the applications. In some embodiments, each of the application windows may correspond to one of the applications. The application windows may be presented on a screen of the mobile device sequentially, simultaneously, substantially simultaneously, or in any other suitable order. In some embodiments, multiple application windows may be presented on the screen of the mobile device based on the order in which their corresponding applications are activated on the mobile device.

[00171] Each of the application windows may be presented in any suitable position of the screen. For example, an application window may be presented in a portion of the screen in which the application window was previously presented. As another example, an application window may be presented in a predetermined portion of the screen (e.g., a portion of the screen defined by one or more coordinates and/or any other positional information). The predetermined portion of the screen may be defined by positional information (e.g., coordinates and/or any other positional information) stored in a configuration file and/or a configuration list associated with an application corresponding to the application window.

[00172] Each of the application windows may have any suitable size. For example, the size of a given application window may be determined based on the type of an application corresponding to the application window (e.g., a phone application, a tablet application, etc.), one or more display parameters of a display of the mobile device (e.g., the display's resolution, pixel density, etc.), and/or any other suitable information. In a more particular

example, process 900 may present an application window of a phone application on a tablet computer by adapting the application window to a size and/or orientation suitable for the display of the tablet computer (e.g., a predetermined width, a predetermined height, etc.). In another more particular example, process 900 may present an application window of a tablet application on a mobile phone by adapting the application window to a size and/or orientation suitable for the display of the mobile phone (e.g., a predetermined width, a predetermined height, etc.).

[00173] In some embodiments, to cause an application window of an application to be presented in a particular size, process 900 may return information about the particular size in response to receiving a request for display parameters made by the application. For example, the application may request a display resolution of the mobile device. Process 900 can return information about the particular size (e.g., a width, a height, etc.) as the display resolution to the application.

[00174] In some embodiments, multiple application windows may be arranged on the screen of the mobile device in any suitable manner. For example, an application window may or may not overlap with one or more portions of the other application windows presented on the screen of the mobile device. In some embodiments, multiple overlapping application windows (e.g., application windows 1110, 1120, and 1130 of FIG. 11) may be arranged on the screen based on priorities associated with the application windows. For example, an application window associated with a relatively higher priority may be positioned on top of an application window associated with a relatively lower priority. In some embodiments, priorities may be assigned to the application windows based on timing information related to the applications (e.g., the order in which the applications are

activated), one or more user interactions with the applications, etc. For example, in some embodiments in which a first application is launched earlier than a second application, a relatively higher priority and a relatively lower priority may be assigned to the first application and the second application, respectively. As another example, a particular priority (e.g., the highest priority, or any other priority) may be assigned to an application window in response to detecting a user interaction with the application window. In some embodiments, the priorities of the application windows may be determined by determining one or more visual orders based on the z-order mechanism described below.

[00175] In some embodiments, process 900 can assign a display status to each of the application windows. A display status corresponding to a given application window may indicate positional information of the application window, visibility of the application window to a user of the mobile device, and/or any other information about the application window. For example, the display status may be a first display status indicating that the application window is presented in the foreground of the screen as the topmost application window (also referred to herein as the "foreground application window"). In such an example, an application process corresponding to the application window may be placed at the top of a stack of application processes. In a more particular example, application window 1110 as illustrated in FIG. 11 may be associated with the first display status. In some embodiments, the first display status may be assigned to multiple application windows presented on the mobile device.

[00176] As another example, the display status may be a second display status indicating that one or more portions of the application window is visible to a user of the mobile device and/or that the given application window is not a foreground application window. In a

more particular example, application windows 1120 and/or 1130 as illustrated in FIG. 11 may be associated with the second display status.

[00177] As still another example, the display status may be a third display status indicating that the given application window is not visible to a user of the mobile device. In some embodiments, upon assigning the third display status to an application window, the application window may be minimized and/or dismissed. Additionally, the application process corresponding to the application may be killed and/or ended by the operating system of the mobile device, when computing resources allocated to the application are to be released.

[00178] While three display statuses are described here, this is merely illustrative. Any suitable number of display statues may be used in accordance with various embodiments of the present disclosure.

[00179] At 915, process 900 can monitor user interactions with one or more of the application windows. One or more of the user interactions may correspond to, for example, a user input that select one or more portions of an application window, a user input that moves an icon or other visual representation of a file from a first application window to a second application window, and/or any other user input. In some embodiments, the user interactions may correspond to one or more user gestures, such as "touch," "press," "swipe," "drag," "double touch," "pinch," and/or any other gesture recognizable by the mobile device.

[00180] At 920, process 900 can present the application windows based on the user interaction(s). For example, process 900 may update the display statuses assigned to the application windows based on the user interaction(s) and can present the application

windows accordingly. More particularly, for example, process 900 may assign the third display status to the application window in response to detecting a user interaction that minimizes and/or dismisses the application window. In a more particular example, the application windows may be rearranged and/or presented by performing one or more operations as described in conjunction with FIG. 9B below.

[00181] As another example, process 900 may rearrange the application windows and/or their corresponding priorities based on the user interactions. More particularly, for example, a higher order may be assigned to an application window in response to detecting a user interaction with the application windows. In some embodiments, the rearranged application window may be presented as a foreground application window. In some embodiments, the application windows may be rearranged by performing one or more operations as described in conjunction with FIG. 10 below.

[00182] FIG. 9B is a flowchart illustrating an example 950 of a process of multi-window application management in accordance with some embodiments of the present disclosure.

[00183] As shown, a first application window for a first application may be opened in step 952. A second application window for a second application may be opened in step 954. For example, the first application window may partially overlap with the second application so that while a user interacts with the first application window, one or more portions of the second application window is visible to the user. The user may also interact with the second application. For example, when the user selects (e.g., clicks, touches, etc.) a portion of the second application window, the second application window may be presented on top of the first application window (e.g., as the foreground application window).

[00184] In step 956, process 950 can determine whether the detected user interaction indicate a user request to minimize the application window. For example, the detected user interaction may be determined as being indicative of a user request to minimize the application window, when the user interaction corresponds to a user selection of a minimization icon or dismissal icon displayed on the application, one or more user gestures (e.g., swiping down the application window, dragging the application window towards a direction, etc.), and/or any other user input requesting the application window be minimized and/or dismissed.

[00185] In response to determining that the detected user interaction corresponds to a user request to minimize the application window, process 950 can proceed to step 958 and can associate a display state with the application window and/or the application corresponding to the application window. For example, in response to detecting a user interaction to minimize the application window (e.g., a selection of a minimization icon), process 950 may associate the application widow with a display status indicating that the given application window is not visible to a user of the mobile device (e.g., the third display status as described in FIG. 9A). In some embodiments, upon assigning the third display status to an application window, the application window may be minimized and/or dismissed. Additionally, the application process corresponding to the application may be killed and/or ended by the operating system of the mobile device, when computing resources allocated to the application are to be released. In some embodiments, onStop() may be called to stop the application window in step 958, and the application process corresponding to the application window may be ended when computing resources (e.g.,

memory, processing resources, etc.) allocated to the application is released by the operating system of the mobile device.

[00186] Alternatively, in response to determining that the detected user interaction does not correspond to a user request to minimize the application window, process 950 can proceed to 960 and can associate the application window with a display status indicating that the application window is visible to a user of the mobile device (e.g., the first display status and/or the second display status as described in connection with FIG. 9A). In some embodiments, onStop() may be prohibited from calling to maintain the application window available on the screen.

[00187] In some embodiments, a visible lifetime constant may be introduced. The visible lifetime constant may define the minimum time for an activity to be changed from an Onstart() status to an Onstop() status so that any activity would be placed in either Onpause() status or Onstart()/Onresume() status before it's changed to an Onstop() status. An activity in this method may maintain all state and member information and remain attached to the window manager for some time before being shut down even if it is completely obscured by new activities.

[00188] In another embodiment, the activity status may be amended so that an activity is set to be in Onstop() status only when it is minimized. Alternatively, the activity may be in Onstart() status when it's placed in foreground. The activity may be in Onpause() status when it's partially or completely covered by other activities.

[00189] According to some embodiments of the present disclosure, multiple application windows presented on the desktop interface may overlap with each other. An order number (also referred to herein as a "visual order") may be assigned to each of the application

windows, so that an application window that is presented (e.g., an application window that is not minimized or dismissed) may be given a distinct order number indicating its corresponding layer on the desktop interface. Multiple order numbers may define how application windows are arranged in the desktop interface. The order numbers may be arranged in an ascent order in some embodiments. For example, a zero order number may be assigned to the desktop interface and/or any other application window and/or interface that is presented at the bottom of the application windows. An application window presented on top of the desktop interface may be associated with a greater order number. The greatest order number may be assigned to one or more application windows presented as the topmost application window(s) (e.g., the foreground application window(s)). Alternatively, the order number may be arranged in a descent order. For example, the zero order number may be assigned to the foreground application window(s). The greatest order number may be assigned to the desktop interface or an application window presented at the bottom of the application windows.

[00190] In some embodiments, the layer of an application window on the desktop interface may also be determined based on a priority associated with an application corresponding to the application window. Multiple application windows may be arranged on the desktop interface based priorities associated with application corresponding to the application windows (e.g., an arrangement of applications in the activity manager 421). A priority associated with an application may or may not correspond to a visual number associated with an application window of the application. In some embodiments, a priority associated with an application and/or a visual order associated with an application window may be adjusted based on a user interaction with the application and/or the application

window. For example, a higher visual order may be assigned to an application window that is originally associated with a lower visual order, in response to a user interaction with the application window.

[00191] More particularly, for example, in the Android™ system, Z-order may be used to define how application windows are arranged, e.g., the order of the application window on the foreground may be 0 and the order of the application window which is right below it may be 1, and so on. In some embodiments, Z-order of application windows may be changed by opening new application window or reopening current existing application windows and the application window in current use may always be order 0. Because in conventional Android™ OS, application windows are always in full-screen mode, the clickable area of the screen is always residing in a particular application window, Z-order may not be changed and the application windows may not be switched by simply clicking on the screens. However, the mobile operating system provided in the present disclosure can switch to the correct application window when the corresponding window is clicked. For example, if there are three windows partially overlapping each other, and the user clicks or taps the window at the bottom, the window at the bottom may be transferred from the bottom to the top, and the focus may be set on it. In some embodiments, a list of Z-order of the current operating application windows and their corresponding areas (e.g., the area bounded by the coordinate of their top-left corners and their bottom-right corners) may be stored. When a user clicks or taps the screen, a touch event may be transmitted to the system together with touch position information that it touches. Then the system may evaluate the first window it hits by analyzing the areas of window with the smallest Z-order which comprises the touch position information. After knowing which window is pointed

to, its Z-value may be changed to 0 and an Onstart() status may be assigned to the corresponding activity. Finally, the Window Manager may be called to refresh and rearrange the application windows so that the application window that is currently pointed to may be moved and presented on the foreground.

[00192] FIG. 10 is a flowchart illustrating a process for arranging application windows for presentation on a mobile device in accordance with some embodiments of the present disclosure. As illustrated, a user activity may be detected in step 1001, the user activity may include a click, a touch, a tag, and/or any other user input indicative of a user interaction with one or more application windows. In step 1002, an application window may be selected based on the user activity detected in step 1001. The application window may or may not be presented on the top of the application windows (e.g., a foreground application window). After the application window is selected, the multi-window manager may be queried in step 1003. In some embodiments, the multi-window manager may be configured to increase the priority of the selected application window (e.g., by querying activity manager and/or window manager). In step 1004, the application windows may be rearranged. For example, in some embodiments in which the application window is not a foreground application window, the application window selected in step 1002 may be placed on the top of the multiple application windows (e.g., being presented as the foreground application window).

[00193] FIG. 11 illustrates an example 1100 of a desktop interface in accordance with some embodiments of the present disclosure. As shown, desktop interface 1100 may include one or more application windows (e.g., application windows 1110, 1120, and 1130), a tool bar 1140, and/or any other element. Tool bar 1140 may include one or more

visual representations of one or more applications (e.g., icons 1141, 1143, 1145, 1147, and 1149). In some embodiments, icons 1141, 1143, 1145, 1147, and 1149 may correspond to applications running on the mobile device. In some embodiments, one or more of icons 1141, 1143, 1145, 1147, and 1149 can correspond to application windows 1110, 1120, and 1130.

[00194] In some embodiments, application windows 1110, 1120, and 1130 may be presented on the desktop interface 1100 sequentially, simultaneously, or in any other order. Application windows 1110, 1120, and 1130 may be arranged and presented on the desktop interface in any suitable manner. For example, each of application windows 1110, 1120, and 1130 may or may not overlap with one or more portions of one or more other application windows. As shown in FIG. 11, application windows 1110 and 1120 may overlap with each other. For example, one or more portions of application window 1110 overlap with the application window 1120. Meanwhile, at least a portion of application window 1110 does not overlap with application window 1120. As such, application window 1110 may also be regarded as being partially overlapping with application window 1120.

[00195] As another example, each of application windows 1110, 1120, and 1130 may have a particular size (e.g., a default size, a size defined by a user, a size determined based on a user interaction with the application window, a recorded size, etc.). The size of application windows 1110, 1120, and 1130 may be adjusted dynamically based on one or more user interactions with one or more of the application windows. Each of the user interactions may correspond to one or more user inputs. For example, one or more of the application windows 1110, 1120, and 1130 may be maximized, minimized, zoomed,

dismissed, etc. based on one or more user gestures (e.g., "touch," "swipe," "pinch," etc.). In some embodiments, one or more application windows may be rearranged within the desktop based on one or more user inputs. For example, a user may move multiple application windows within the desktop interface using one or more gestures (e.g., "drag" and/or "drop" multiple application windows using two or more fingers of a user).

[00196] In some embodiments, before a new application window of an application is presented on the desktop interface, a test can be conducted to determine if any application window of the application has been previously presented on the desktop interface. If no application window is opened, the new application window may be displayed on the desktop interface based on a default position and/or a default size. If one or more application windows of the application were previously presented on the desktop interface, the new application window can be presented on the desktop interface based on the position of an application window that was previously presented for the application (e.g., the most recently presented application window).

[00197] FIG. 12 is a flowchart illustrating a method for arranging application windows in accordance with some embodiments of the present disclosure.

[00198] In step 1201, an application window of an application may be requested to display. In some embodiments, the icon of the application may be either touched or clicked.

[00199] In accordance with step 1201, the application configuration list can be queried in step 1202. The application configuration list is described elsewhere in the present disclosure.

[00200] In step 1203, it can be determined that if the application is on the application configuration list queried in step 1202. If the application is not on the application

configuration list, step 1204 can be performed and the default size of the application window can be extracted. If the application is on the application configuration list, the size of the application window stored in the application configuration list can be extracted in step 1205.

[00201] The multi-window manager can be queried in step 1206 for determining the position of the application window. In step 1207, the multi-window manager can determine if any application has been opened on a desktop interface (e.g., the desktop interface 1100 as described in connection with FIG. 11). In response to determining that one or more applications have been opened on the desktop interface, step 1208 can be performed and the coordinate of the application window may be calibrated based on the default coordinate, or on the coordinate of the application window stored in the application configuration list or the application type list. The calibration may include a shift, or the like. If no application is opened on the desktop, the default coordinate or the coordinate of the application window stored in the application configuration list or the application type list can be extracted in step 1209.

[00202] In step 1210, the application window can be opened and presented on the desktop interface. Additionally or alternatively, a visual representation of the application (e.g., an icon corresponding to the application) can be placed on a task bar of the desktop interface.

[00203] It should be noted that the flowchart described above is provided for the purposes of illustration, and not intended to limit the scope of the present disclosure. For persons having ordinary skills in the art, a certain amount of variations and modifications may be conceptualized and reduced to practice in the light of the present disclosure.

However, those variations and modifications may not depart from the protecting scope of the present disclosure.

[00204] FIG. 13 is a diagram illustrating a mechanism for running an application on various mobile devices in accordance with some embodiments of the present disclosure. Referring to FIG. 13, an application window 1310a corresponding to a phone application can be displayed on a screen of a mobile phone 1320. In some embodiments, application window 1310a may be presented in a full-screen mode (e.g., encompassing the whole screen of the mobile phone). When the phone application is opened in a tablet computer 1330, an application window 1310b corresponding to the phone application can be presented on a screen of the tablet computer. As shown in FIG. 13, application windows 1310a and 1310b may have the same size (e.g., the same height and/or width). The application windows 1310a and 1310b may be the same in some embodiments. As such, a user can open a phone application in the tablet computer 1330 without rotating the tablet computer 1330 to view user interfaces provided by the application. In some embodiments, when a tablet application is opened in the tablet computer 1330, an application window of the tablet application can be presented on the screen of the tablet computer in full screen mode, either with or without the task bar.

[00205] FIG. 14 is a flowchart illustrating an example 1400 of a method for presenting application windows of an application on various mobile devices in accordance with some embodiments of the present disclosure. An application can be opened in step 1401. In step 1402, the type of the application can be detected. A type of the application may be determined in step 1403 by querying the application type list. For example, the application can be determined as a phone application or a tablet application, as shown in blocks 1404

and 1405. One or more application windows of the application can be opened based on the type of the application. For example, if the application is determined to be a phone application, one or more application windows of the application may be presented in a phone resolution (e.g., a predetermined size defined by a width and a height). In a more particular example, an application window of a phone application may be presented on a mobile phone (e.g., a full-screen mode as illustrated in FIG. 13). Alternatively, an application window of the phone application may be presented on a tablet computer in the phone resolution (e.g., application window 1310b as described in connection with FIG. 13 above). Exemplary phone resolutions may include 1536 x 1152, 1920 x 1152, 1920 x 1080 1920 x 1200, 2048 x 1536, 2560 x 1536, 2560 x 1600, or the like. If the application is a tablet application, an application window of the tablet application may be presented on a mobile device in a predetermined size associated with the tablet application in step 1407. For example, the application window may be presented in a full-screen mode on the mobile device (e.g., a mobile phone, a tablet computer, etc.).

[00206] It should be noted that the flowchart described above is provided for the purposes of illustration, and not intended to limit the scope of the present disclosure. For persons having ordinary skills in the art, various modifications and variations may be conducted under the teachings of the present disclosure. However, those modifications and variations may not depart from the scope of the present disclosure. For example, step 1402 and step 1403 may be merged into a single step, the type of an application may be detected by querying the application configuration list or the application type list as described elsewhere in the present disclosure.

[00207] FIG. 15 illustrates a drag and drop operation that can move an object across multiple application windows in accordance with some embodiments of the present disclosure.

[00208] As shown, application windows 1510 and 1520 may be presented on a desktop interface 1500. An object 1521 located in application window 1520 can be moved to application window 1510 by a user. The object 1521 may then be presented in application window 1510. For example, a user may drag the object 1521 from application window 1520 to application 1510. When the user releases (e.g., drops) the object 1521 in a particular position within application window 1510, the object 1521 may be presented in application window 1510 (e.g., in the particular position). In some embodiments, the object 1521 may correspond to an icon of a file, an image, text, and/or any other information presented in application window 1520.

[00209] In some embodiments, an application window may be divided into a certain number of regions, some of the regions may be configured to accept files from other application windows, the rest regions may be configured to reject files from other application windows. When the object 1521 is released in a region (or a position) that is incapable of receiving objects, an indicator may be displayed on the desktop interface 1500 to warn the user. When the object is released in a region that is capable of receiving objects, another indicator may be displayed on the screen to indicate that the object 1521 can be dropped in the region. In some embodiments, the object 1521 may be moved from application 1520 to application window 1510 by performing one or more operations described below in connection with FIG. 16.

[00210] FIG. 16 is a flowchart illustrating an example 1600 of a process for moving an object across multiple application windows in accordance with some embodiments of the disclosure.

[00211] As illustrated, process 1600 may begin by receiving a user selection of an object presented in a first application window in step 1601. The object may be any object that can be moved (e.g., dragged and dropped) from the first application to one or more other application windows. The object may correspond to an icon of a file, an image, text, and/or any other information presented in the first application window.

[00212] In step 1602, process 1600 may detect a user input that moves the object towards a second application window. For example, the user input may correspond to a "drag" gesture and/or any other user input that indicates a user request to move the object. More particularly, for example, process 1600 may determine that the object is moved towards the second application window upon detecting that one or more portions of the object have been placed within the second application window. In some embodiments, process 1600 may determine positional information (e.g., one or more coordinates) of the object. Additionally, process 1600 may send the positional information to a second application corresponding to the second application window.

[00213] In some embodiments, when the object is moved (e.g., dragged), process 1600 may determine one or more features of the object. For example, process 1600 may determine a type of the object, such as an MIME (Multipurpose Internet Mail Extensions) type of the object. In some embodiments, one or more of the features may be determined from a first application that corresponds to the first application window. Additionally,

process 1600 may send information about the features (e.g., the type of the object) to the second application.

[00214] In step 1604, process 1600 may determine whether the object has been successfully moved into a second application window. For example, process 1600 may determine that the object has been successfully moved into the second application window upon detecting a user input that releases the object in any portion of the second application window. Such a user input may be and/or include a user input that drops the object within the second apparition window (e.g., a successful "drop" gesture).

[00215] In some embodiments, in response to determining that the object has been moved into the second application window, process 1600 may proceed to step 1605 and may further determine whether a second application corresponding to the second application window is capable of receiving the object. This determination can be made, for example, based on the features obtained in step 1603. For example, process 1600 can determine that the second application is capable of receiving the object in response to determining that the second application is capable of accepting a drag and drop event of an object of the determined type of the object (e.g., the MIME type). In some embodiments, the type of the data may be used by the second application to determine whether the second application can place the object in a particular position in the second application window (e.g., a position in which the user release the object).

[00216] In some embodiments, in response to determining that the second application is capable of receiving the object (e.g., placing the object in the position in which the user releases the object), process 1600 can proceed to step 1606 and can place the object in the second application window. Alternatively, in response to determining that the second

application is not capable of receiving the object, process 1600 may place the object on a desktop interface in which the first application window and the second window are presented.

[00217] In some embodiments, in response to determining that the object has not been successfully moved to the second application (e.g., "NO" at step 1604), process 1600 may proceed to step 1607 and may place the object on the desktop interface. In some embodiments, process 1600 can determine that the object has not been successfully moved to the second application upon detecting an unsuccessful "drop" operation (e.g., detecting that the object has not been moved to the second application window when the user releases the object).

[00218] In some embodiments, a notification may be sent to the first application and/or the second application, respectively. The notification may indicate that the object has been successfully moved into the second application window (e.g., upon detecting a successful "drag and drop" event). In some embodiments, the first application and/or the second application may perform further actions based on the notification. For example, the first application may update the first application window to provide an application window without a visual representation of the object. As another example, the second application may update the second application by presenting a visual representation of the object in the second application window. In some embodiments, an error message may be prompted to indicate an unsuccessful "drag and drop" event to the first application and/or the second application.

[00219] It should be noted that the above steps of the flow diagrams of FIGS. 9A, 9B, 10, 12, 14, and 16 can be executed or performed in any order or sequence not limited to

the order and sequence shown and described in the figure. Also, some of the above steps of the flow diagrams of FIGS. 9A, 9B, 10, 12, 14, and 16 can be executed or performed substantially simultaneously where appropriate or in parallel to reduce latency and processing times. Furthermore, it should be noted that FIGS. 9A, 9B, 10, 12, 14, and 16 are provided as examples only. At least some of the steps shown in the figures may be performed in a different order than represented, performed concurrently, or altogether omitted. In some embodiments, one or more operations described in connection with FIGS. 9A, 9B, 10, 12, 14, and 16 may be perform by one or more processors, such as a processor executing the multi-window manager described in the present disclosure.

[00220] FIG. 17 is a user interface that may be employed by the mobile device 100 in accordance with some embodiments of the present disclosure. Referring to FIG. 17, the user interface 1700 can include a start menu button 1710, a task bar 1720, and a desktop 1740. Shortcuts of application may be placed on the desktop 1740 as illustrated in the figure. The start menu button 1710 can be configured to receive a user command, the user command can include a mouse click, a finger touch, a voice control command, and the like. Once the start menu button 1710 is activated, a start menu can be displayed on the user interface that is described elsewhere in the present disclosure. The task bar 1720 can be configured to display a number of shortcuts of applications. Meanwhile, when an application is opened by a user, the icon corresponding to the application can be displayed on the task bar. As illustrated in the figure, as the application 1730 is opened, the application window 1731 for the application 1730 is displayed in the desktop 1740. The application icon 1732 of the application 1730 is displayed on the task bar 1720. In the right

area of the task bar 1720, a number of setting options may be displayed. The setting options can include voice volume, network, notification center, time, date, and the like.

[00221] FIG. 18 illustrates a start menu in accordance with one embodiment of the present disclosure. Once the start menu button 1710 is activated, start menu 1900 can be opened as illustrated in FIG. 18. Start menu 1900 can include shortcut region 1910 and application region 1920. Shortcut region 1910 can include a user setting icon 1950 and a number of shortcut of settings. By clicking user setting region 1950, a user may be able to set the profile of a user. Application region 1920 can include all icons of applications that are installed on the mobile device 100. Search box 1940 can be used to search applications installed on the mobile device 100 by typing the name of an application. Arrow 1930 can be used to maximize start menu 1900 and a full screen mode can be displayed.

[00222] FIG. 19 illustrates a start menu in full screen mode in accordance with another embodiment of the present disclosure. The details of start menu 1900 may refer to the description provided in FIG. 18.

[00223] FIG. 20 is block diagram illustrating a file manager that may be employed in the mobile device 100 in accordance with some embodiments of the present disclosure. Referring to FIG. 5, file manager 2000 can include a tool bar 2010, a text region 2020, a path bar 2030, an edit bar 2040, and a sidebar 2050. Tool bar 2010 can be configured to perform functions including backward, forward, refreshing, changing display mode, settings, searching file, Maximization, minimization, closing. Text region 2020 can be configured to display files contained in a folder. Path bar 2030 can be configured to display current path. Edit bar 2040 can be configured to edit files. In some embodiments, edit bar

2040 may be activated after a long press of a file in a folder. Sidebar 2050 can be configured to display frequently used folders and configure network.

[00224] FIG. 21 is a block diagram illustrating a notification center that may be employed in the mobile device 100 in accordance with some embodiments of the present disclosure. Region 2110 can be configured to display time and date. Button 2160 can be configured to manage application notifications. In some embodiments, when button 2160 is clicked or touched, an application window for application notification (not shown in the figure) may be popped out for users to manage application notifications. At least partial of all applications installed on the mobile device 100 can be displayed in the application window for application notification. A user may configure priority of applications listed in the application window, or choose to block one or more application notifications. Region 2120 can be configured to display all application notifications. Button 2130 can be configured to clear notification center. Region 2140 can be configured to display system settings including Wi-Fi, Bluetooth™, GPS, flight mode, mute, tablet mode, automatic rotation, screen shots, or the like. Control bar 2150 can be configured to control the brightness of screen, for example, the display 110.

[00225] FIG. 22 is a block diagram illustrating a security center that may be employed in the mobile device 100 in accordance with some embodiments of the present disclosure. Referring to the FIG. 22, the security center may be displayed on a screen as an application window. The security center 2250 can include self-start permission 2210, power usage 2220, location information 2230, and application management 2240. Self-start permission 2210 can be configured to prohibit applications from starting automatically. Power usage 2220 can be configured to display power usage. Location information 2230 can be

configured to turn on or turn off location information. Application management 2240 can be configured to manage at least partial of all applications installed on the mobile device 100.

[00226] FIG. 23 illustrates an example 2300 of a user interface for implementing a file manager in a mobile device in accordance with some embodiments of the present disclosure. As shown, user interface (UI) 2300 can include an application window 2005 for implementing a file manager on the mobile device. The application window 2005 may include a text region 2020, a path bar 2030, an edit bar 2040, a sidebar 2050, an adjustment button 2310, and/or any other UI element for implementing a file manager on the mobile device.

[00227] The text region 2020 can be configured to display one or more virtual representations of one or more files contained in a folder. Each of the virtual representations may include graphics, images, text, and/or any other content that may be used to represent a file. The path bar 2030 can be configured to display a current path. The path may represent the directory relationship in a file system and indicate the directory of one or more files or folders in the file system. The path may be expressed by a string of characters in which each directory separated by a delimiting character, for example, a slash or a backslash. Directory may include parent directory, current directory, subdirectory, etc. The edit bar 2040 can be configured to provide one or more edit functions that enable a user of the mobile device to edit one or more of the files, the virtual representations of the files, and/or any other information about the files. Examples of the edit functions may include "copy," "paste," "cut," "link," "delete," etc. The edit bar 2040 may include one or more UI elements (e.g., one or more buttons, icons, etc.) corresponding to the edit functions.

A user can cause one or more of the edit functions to be performed by the mobile device by interacting with one or more portions (e.g., UI elements) of the edit bar 2040. The sidebar 2050 can be configured to display information about frequently used folders and network configuration. The sidebar 2050 can also provide functionality that enables a user to configure network settings, edit the frequently used folders, etc.

[00228] The adjustment button 2310 can provide functionality that enables a user to resize the application window 2005 of the file manager. The mobile device 100 may detect a user interaction with the adjustment button 2310 and may then enlarge or reduce the size of the application window 2005 of the file manager to any size the user desires. As will be discussed in more detail below, the user interaction may be made via one or more user inputs that may include and/or correspond to one or more gestures of a user of the mobile device 100, such as "touch," "press," "swipe," "drag," "double touch," "pinch," and/or any other gesture recognized by the mobile device 100. In some embodiments, the user may press and hold a mouse button (e.g., a left button, a right button, etc.) and then move the adjustment button 2310 to resize the application window 2005 of file manager. In some embodiments, the user may use their finger or stylus on the adjustment button 2310 and move the adjustment button 2310 in any direction to resize the application window 2005 of the file manager. In some embodiments, the user may single click the adjustment button 2310 to enlarge the application window 2005 of the file manager proportionally, and double click the adjustment button 2310 to reduce the application window 2005 of the file manager proportionally. As another example, the user may single click the adjustment button 2310 to reduce the application window 2005 of the file manager proportionally, and double click the adjustment button 2310 to enlarge the application window 2005 of the file

manager proportionally. In some embodiments, to resize the application window 2005 of the file manager, the user may place two fingers on the application window 2005 of the file manager and swipe the two fingers (e.g., in opposite directions). The user may also keep one finger in a fixed position while swiping another. In some embodiments, the user may single click the adjustment button 2310 and the application window 2005 of the file manager may be resized by moving the cursor of a mouse in any direction while keeping the left button or the right button of the mouse pressed.

[00229] The adjustment button 2310 may be located in any portion of the application window 2005, such as one or more of tool bar 2010, text region 2020, path bar 2030, edit bar 2040, sidebar 2050, or any other portion of the application window 2005. It should be noted that in FIG. 23, the file manager is a particular example of application. In some embodiments, more applications, such as a web browser application, a game application, a financial portal application, an electronic book application or any application installed on the mobile device 100, may be employed with the adjustment button 2310 for resizing the application window.

[00230] FIG. 24 illustrates an example 2400 of a user interface for implementing page-switching on a mobile device in accordance with some embodiments of the present disclosure. As shown, user interface 2400 may include one or more application windows 2410. Each of application windows may correspond to one of the applications, such as a web browser application, a news application, an electronic book application or any other application installed on the mobile device. Application window 2410 can include one or more display areas 2415 for presenting content, a page-switching indicator 2420, a

page-switching button 2430, and/or any other UI element for implementing page-switching functionality on a mobile device.

[00231] Display area 2415 can be configured to display any suitable content, such as text, one or more images, graphics, video content, etc. The content displayed in display area 2415 may correspond to an article, a news item, a webpage, one or more portions of an electronic book, a group of images, etc. In some embodiments, the content may be divided into a number of pages based on the size of display (e.g., the size of the application window 2410, the size of the display area 2415, the display resolution of the mobile device, etc.). For example, as shown in FIG. 24, the content may be divided into pages 2417a, 2417b, etc. Each of the pages may be displayed in the display area 2415. Multiple pages of the content may or may not include the same content.

[00232] The content may be partially displayed in display area 2415 based on the size of the display area, the size of the application window, and/or the display resolution of the mobile device. For example, page 2417a of the content may be displayed in the display area 2415. The user can access one or more other portions and/or all of the content (e.g., page 2417b and/or any other page of the content) by interacting with the application window 2410. For example, the user can scroll the content (e.g., page 2417a) displayed in the display area 2415 to cause one or more other portions of the content (e.g., page 2417b) to be displayed in the display area 2415. As another example, the user can select (e.g., click, press, tap, etc.) the page-switching button 2430 to scroll the content displayed in the display area 2415.

[00233] In some embodiments, in response to detecting a user interaction indicative of a user request to access another page or another portion of the content (e.g., a user selection

of the page-switching button 2430, a gesture that scrolls down the page displayed in the display area 2415, etc.), the mobile device can cause the content to be moved (e.g., scrolled down) in the display area 2415 and can display the page-switching indicator 2420 on the application window 2410 along with the moving content. When the content stops moving (e.g., when the next page 2417b has been displayed in the display area 2415), the page-switching indicator 2420 may disappear.

[00234] For example, as illustrated in FIG. 24, a user may select (e.g., press, click, tap, etc.) the page-switching button 2430 to access page 2417b of the content. The content (e.g., page 2417a) displayed in the display area 2015 may move in the display area 2015. While the page-switching button 2430 is selected, the page-switching indicator 2420 may be displayed. When the content moves, the page-switching indicator 2420 may be displayed in the bottom of the application window until the content stops moving, and a new page is displayed. In some embodiments, the page-switching indicator 2420 may move simultaneously while the content is moving. In some embodiments, the page-switching indicator 2420 may disappear gradually after the content stops moving. For example, the page-switching indicator 2420 may last for a few seconds or any other period of time after the content stop moving.

[00235] In some embodiments, the page-switching button 2430 may be selected by one or more of the user inputs that may include and/or correspond to one or more gestures of a user of the mobile device 100, such as "touch," "press," "swipe," "drag," "double touch," "pinch," and/or any other gesture recognizable by the mobile device 100. In some embodiments, the page-switching button 2430 can be located in any area of the application window 2410, such as the top, the left, the right, or the like. In some embodiment, the

page-switching indicator 2420 may be displayed in any part of the application window 2410. For example, when the content is moving backward (e.g., moving to a previous page), the page-switching indicator 2420 may be displayed on the top of the application window 2410. As another example, the page-switching indicator 2420 may be displayed in the left part of the application window, or the right part of the application window.

[00236] In some embodiments, one or more functions of the page-switching button 2430 may be defined by one or more user inputs. For example, in response to receiving a user input that indicates a user request to set the function(s) (e.g., long pressing the page-switching button 2430), a configuration menu (not shown in FIG. 24) may be presented on the application window 2410 to prompt the user to set one or more functions of the page-switching button 2430 (e.g., page up, page down, page left, page right, or the like in response to a user interaction with the page-switching button 2430).

[00237] In some embodiments, the user may move (e.g., drag) the page-switching button 2430 towards a direction (e.g., up, down, left, right, etc.) to move the content displayed in the display area 2415. The page-switching button 2430 may be moved using a user's one or more fingers, a stylus, etc. In some embodiments, the page-switching button 2430 may be divided into multiple parts, each part representing a corresponding direction (e.g., up, down, left, right, etc.) in which the content may be moved. The user can select the different parts to move the content in their corresponding directions. In some embodiments, the page-switching button 2430 may include one or more hidden buttons, such as a page-up button, a page-down button, a page-left button, a page-right button, etc. The hidden button(s) may be popped out when the page-switching button 2430 is selected. The user may press one or more of the hidden buttons to move the content in different directions

corresponding to the hidden buttons. In some embodiments, the user may scroll down the content by single clicking the page-switching button 2430, and may scroll up the content by double clicking the page-switching button 2430. The user may also scroll down the content by double clicking the page-switching button 2430, and may scroll up the content by single clicking the page-switching button 2430. In some embodiments, the user may scroll the content towards a direction in the application window 2410 (e.g., up, down, left, right, etc.) using one or more fingers, a stylus, and/or any other input device. In some embodiments, the user may scroll the content by pressing a button on a keyboard, such as page up and page down keys, clicking a mouse on a specific area of the application window 2410 (e.g., the left bottom corner, the right bottom corner of the application window 2410, etc.), etc. In some embodiments, the user may scroll the content by pressing and/or rotating one or more physical buttons mounted on the mobile device. In some embodiments, the displayed contents can be scrolled in any direction, such as backward scrolling, forward scrolling, leftward scrolling, rightward scrolling, diagonal scrolling, etc.

[00238] In some embodiments, the page-switching indicator 2420 may be generated and displayed when the page-switching button 2430 is initiated to facilitate a user's consumption of content presented in the application window. For example, when the content displayed in the application window (e.g., a page of the content) is under moving operation, the user may lose their reading focus and may have to relocate the content to resume reading. The page-switching indicator 2420 may be configured to indicate a connection part of two portions of the content displayed in the application window (e.g., two pages) to enable the user to relocate the content properly. For example, if the displayed content is an electronic text, the connection part of two successive pages of content may

be a boundary portion between the last row of text that is displayed on a first page and the first row of a text that is displayed on a second page. The connection part may also include other text around the boundary portion. As another example, if the displayed content is an electronic image, the connection part may be and/or include a boundary portion between the end portion of the image displayed on a previous view and the header portion of the image displayed on a next view. The connection part may also include other image around the boundary portion. In some embodiments, the connection part may be a boundary portion between the header portion of the previous view and the end portion of the next view. In some embodiments, the connection part may be a boundary portion between the rightmost portion of the previous view and the leftmost portion of the next view. Alternatively or additionally, the connection part may be a boundary portion between the leftmost portion of the front view and the rightmost portion of the next view.

[00239] In some embodiments, the page-switching indicator 2420 may remain on the application window 2410 after the scrolling operation ends. Alternatively or additionally, the page-switching indicator 2420 may be dismissed when the content stops moving. Further, the page-switching indicator 2420 may be dismissed immediately or after a predetermined time period. The page-switching indicator 2420 may also be reset or regenerated at a new connection part after each new page is displayed during continuous scrolling operations (e.g., scrolling up or down multiple pages of the content).

[00240] The page-switching indicator 2420 may be implemented in different forms, such as lines, arrows, balloons, symbols, and/or any other UI element of any shape and can be placed at any area of the connecting part. In some embodiments, any application running on the mobile device (e.g., an electronic book application, a news application, a

message application or any other application) can utilize the page-switching button and/or the page-switching indicator to perform one or more functions described above. In some embodiments, the page-switching button 2430 may be implemented as one or more physical buttons mounted on the mobile device 100.

[00241] FIG. 25 illustrates an example of mechanisms for user interface swap according to some embodiments of the present disclosure. As shown, each of a user interface 2500 and the user interface 1700 as described in connection with FIG. 17 may be presented on the mobile device 100. User interfaces 1700 and 2500 may be swapped in response to one or more user inputs indicative of a user request to swap between the user interface 1700 and the user interface 2500.

[00242] In some embodiments, a user may initiate a swap of user interfaces 1700 and 2500 by selecting (e.g., clicking, touching, pressing, rotating, etc.) a switch for user interface swap. For example, when the switch is turned on, the multi-window display mode provided by the mobile device 100 may be turned on and the user interface 1700 may be displayed. When the switch is turned off, the multi-window display mode provided by the mobile device 100 may be turned off and the user interface 2500 may be displayed. In some embodiments, the switch may be a soft key displayed on the mobile device 100, a physical key (e.g., a key that may be pressed, rotated, etc.), etc.

[00243] In some embodiments, a user may touch a soft key displayed on the interface 1700 to initiate a switch from the user interface 1700 to the user interface 2500. The soft key may be displayed on the application region 1920 of the start menu 1900 as described in connection with FIG. 18. The soft key may correspond to an application whose icon is displayed in the application region 1920. In some embodiments, the soft key may be set

in the shortcut region 1910 as described in connection with FIG. 18. In some embodiments, the soft key may correspond to an application whose icon is displayed in the user interface 2500. A user may initiate a swap from the user interface 2500 to the user interface 1700 by selecting (e.g., clicking, touching, etc.) the soft key. In some embodiments, bidirectional swaps between the user interface 1700 and the user interface 2500 may be initiated by manipulating a keyboard. Shortcuts of the keyboard may be self-defined by a user to initiate the swap.

[00244] In some embodiments, the user interface 2500 may be interface of an Android™ OS.

[00245] It should be noted that the description of the user interface swap is provided for the purposes of illustration, and not intended to limit the scope of the present disclosure. For persons having ordinary skills in the art, various modifications and variations may be conducted under the teaching of the present disclosure. However, those modifications and variations may not depart from the scope of the present disclosure. For example, after the switch is turned on, the user interface 2500 may be displayed, while the switch is turned off, the user interface 1700 may be displayed.

[00246] FIG. 26 illustrates an example 2600 of a user interface for implementing a file manager in the mobile device in accordance with some embodiments of the present disclosure. As shown, user interface (UI) 2600 can include an application window 2620 for implementing a file manager on the mobile device. The application window 2620 may include a text region 2020, a path bar 2030, an edit bar 2040, a sidebar 2050, and an adjustment button 2610, and/or any other UI element for implementing a file manager on the mobile device.

[00247] The text region 2020 can be configured to display one or more virtual representations of one or more files contained in a folder. Each of the virtual representations may include graphics, images, text, and/or any other content that may be used to represent a file. The path bar 2030 can be configured to display a current path. The path may represent the directory relationship in a file system and indicate the directory of one or more files or folders in the file system. The path may be expressed by a string of characters in which each directory is separated by a delimiting character (e.g., a slash, a backslash, etc.). A directory may include a parent directory, current directory, subdirectory, etc. The edit bar 2040 can be configured to provide one or more edit functions that enable a user of the mobile device to edit one or more of the files, the virtual representations of the files, and/or any other information about the files. Examples of the edit functions may include "copy," "paste," "cut," "link," "delete," etc. The edit bar 2040 may include one or more UI elements (e.g., one or more buttons, icons, etc.) corresponding to the edit functions. A user can cause one or more of the edit functions to be performed by the mobile device by interacting with one or more portions (e.g., UI elements) of the edit bar 2040. The sidebar 2050 can be configured to display information about frequently used folders and network configuration. The sidebar 2050 can also provide functionality that enables a user to configure network settings, edit the frequently used folders, etc.

[00248] The adjustment button 2610 can provide functionality that enables a user to resize the application window 2620 of the file manager. The mobile device 100 may detect a user interaction with the adjustment button 2610 and may then adjust the application window 2620 (e.g., by enlarging, reducing, etc. the application window 2620 to any size the user desired, by maximizing the application window 2620, etc.) based on the user

interaction. As will be discussed in more detail below, the user interaction may be made via one or more user inputs that may include and/or correspond to one or more gestures of a user of the mobile device 100, such as "touch," "press," "swipe," "drag," "double touch," "pinch," and/or any other gesture recognized by the mobile device 100. In some embodiments, the user may single click the adjustment button 2610 to maximize the application window 2620 of the file manager to fill the entire screen (full screen mode), and double click the adjustment button 2610 to minimize the application window 2620 of the file manager. As another example, the user may double click the adjustment button 2610 to maximize the application window 2620 of the file manager to fill the entire screen, and single click the adjustment button 2610 to minimize the application window 2620 of the file manager. As a further example, by default, the mobile device may rearrange the minimized application window under a title mode or reduce the minimized application window to the size of a taskbar button. In some embodiments, the mobile device can resize the application window 2620 in response to detecting a user interaction that selects and/or moves the adjustment button 2610. For example, the user may press and hold a mouse button (e.g., a left button, a right button, etc.) and then drag the adjustment button 2610 to resize the application window 2620 of the file manager. In some embodiments, the user may select the adjustment button 2610 using one or more fingers or a stylus and may move the adjustment button 2610 in any direction to resize the application window 2620 of the file manager. In some embodiments, when the adjustment button 2610 is dragged to reach an edge of the screen, the mobile device may maximize the application window 2620 of the file manager to full screen mode automatically (e.g., by adjusting the application window 2620 to encompass the entire screen of the mobile device). In some embodiments,

to resize the application window 2620 of the file manager, the user may place two fingers on the application window 2620 of the file manager and swipe the two fingers (e.g., in opposite directions). The user may also keep one finger in a fixed position while swiping another. In some embodiments, the user may single click the adjustment button 2610 and the application window 2620 of the file manager may be resized while the user moves the cursor of a mouse in any direction and keeps the left button or the right button of the mouse pressed.

[00249] The adjustment button 2610 may be located in any portion of the application window 2620, such as one or more of tool bar 2010, text region 2020, path bar 2030, edit bar 2040, sidebar 2050, or any other portion of the application window 2620. It should be noted that in FIG. 26, the file manager is a particular example of application. In some embodiments, more applications, such as a web browser application, a game application, a financial portal application, an electronic book application or any application installed on the mobile device 100, may be employed with the adjustment button 2610 for resizing the application window.

[00250] Having thus described the basic concepts, it may be rather apparent to those skilled in the art after reading this detailed disclosure that the foregoing detailed disclosure is intended to be presented by way of example only and is not limiting. Various alterations, improvements, and modifications may occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested by this disclosure, and are within the spirit and scope of the exemplary embodiments of this disclosure.

[00251] Moreover, certain terminology has been used to describe embodiments of the present disclosure. For example, the terms “one embodiment,” “an embodiment,” and/or “some embodiments” mean that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the present disclosure. Therefore, it is emphasized and should be appreciated that two or more references to “an embodiment” or “one embodiment” or “an alternative embodiment” in various portions of this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures or characteristics may be combined as suitable in one or more embodiments of the present disclosure.

[00252] Further, it will be appreciated by one skilled in the art, aspects of the present disclosure may be illustrated and described herein in any of a number of patentable classes or context including any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof. Accordingly, aspects of the present disclosure may be implemented entirely hardware, entirely software (including firmware, resident software, micro-code, etc.) or combining software and hardware implementation that may all generally be referred to herein as a "block," “module,” “engine,” “unit,” "component," or "system." Furthermore, aspects of the present disclosure may take the form of a computer program product embodied in one or more computer readable media having computer readable program code embodied thereon.

[00253] The terms “first,” “second,” “third,” “fourth,” etc. as used herein are meant as labels to distinguish among different elements and may not necessarily have an ordinal meaning according to their numerical designation.

[00254] The disclosure also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a machine readable storage medium, such as, but not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, each coupled to a computer system bus.

[00255] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the method steps. The structure for a variety of these systems will appear as set forth in the description below. In addition, the disclosure is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the disclosure as described herein.

[00256] In some embodiments, any suitable computer readable media can be used for storing instructions for performing the processes described herein. For example, in some embodiments, computer readable media can be transitory or non-transitory. For example, non-transitory computer readable media can include media such as magnetic media (such as hard disks, floppy disks, and/or any other suitable media), optical media (such as compact discs, digital video discs, Blu-ray discs, and/or any other suitable optical media), semiconductor media (such as flash memory, electrically programmable read only

memory (EPROM), electrically erasable programmable read only memory (EEPROM), and/or any other suitable semiconductor media), any suitable media that is not fleeting or devoid of any semblance of permanence during transmission, and/or any suitable tangible media. As another example, transitory computer readable media can include signals on networks, in wires, conductors, optical fibers, circuits, any suitable media that is fleeting and devoid of any semblance of permanence during transmission, and/or any suitable intangible media.

[00257] Computer program code for carrying out operations for aspects of the present disclosure may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Scala, Smalltalk, Eiffel, JADE, Emerald, C++, C#, VB. NET, Python or the like, conventional procedural programming languages, such as the "C" programming language, Visual Basic, Fortran 2003, Perl, COBOL 2002, PHP, ABAP, dynamic programming languages such as Python, Ruby and Groovy, or other programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider) or in a cloud computing environment or offered as a service such as a Software as a Service (SaaS).

[00258] Furthermore, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations therefore, is not intended to limit the claimed

processes and methods to any order except as may be specified in the claims. Although the above disclosure discusses through various examples what is currently considered to be a variety of useful embodiments of the disclosure, it is to be understood that such detail is solely for that purpose, and that the appended claims are not limited to the disclosed embodiments, but, on the contrary, are intended to cover modifications and equivalent arrangements that are within the spirit and scope of the disclosed embodiments. For example, although the implementation of various components described above may be embodied in a hardware device, it may also be implemented as a software only solution—e.g., an installation on an existing server or mobile device.

[00259] Reference throughout this specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment. Thus, the appearances of the phrase "in one embodiment" or "in an embodiment" in various places throughout this specification are not necessarily all referring to the same embodiment. In addition, the term "or" is intended to mean an inclusive "or" rather than an exclusive "or."

[00260] Similarly, it should be appreciated that in the foregoing description of embodiments of the present disclosure, various features are sometimes grouped together in a single embodiment, figure, or description thereof for the purpose of streamlining the disclosure aiding in the understanding of one or more of the various inventive embodiments. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed subject matter requires more features than are expressly recited in each claim. Rather, inventive embodiments lie in less than all features of a single foregoing disclosed embodiment.

WHAT IS CLAIMED IS:

1. A method for providing a mobile operating system, the method comprising:

displaying a desktop interface on a screen of a mobile device; and

displaying a first application window and a second application window on the desktop interface, wherein the first application window corresponds to a first application running on the mobile device, wherein the second application window corresponds to a second application running on the mobile device, wherein at least a portion of the first application window does not overlap with the second application window; and

wherein the first application window comprises an adjustment button configured to adjust the size of the first application window and to maximize the first application window.

2. The method of claim 1, further comprising displaying a task bar on the desktop interface, wherein the taskbar comprises a first visual representation of the first application and a second visual representation of the second application.

3. The method of claim 2, wherein the desktop interface further comprises a start menu button for displaying a start menu.

4. The method of claim 1, wherein displaying the first application window and the second application window on the desktop interface further comprises:

querying an application configuration list for a first size and a first coordinate; and

displaying the first application window based on the first size and the first coordinate.

5. The method of claim 4, wherein displaying the first application window and the second application window on the desktop interface further comprises:

querying the application configuration list for a second size and a second coordinate; and

displaying the second application window based on the second size and the second coordinate.

6. The method of claim 4, wherein displaying the first application window and the second application window on the desktop interface further comprises:

querying the application configuration list for extracting a third size;

calculating, based on the first coordinate, a third coordinate; and

displaying the second application window based on the third size and the third coordinate.

7. The method of claim 4, wherein the second size corresponds to a type of the second application.

8. The method of claim 7, wherein the type of the second application is one of a phone application and a tablet application.

9. The method of claim 1, wherein at least a portion of the first application window overlaps with the second application window.

10. The method of claim 9, further comprising:

detecting a user interaction with at least one of the first application window and the second

application window; and

rearranging the first application window and the second application window based on the user interaction, wherein the rearranged first application window is partially overlapped with the second application window.

11. The method of claim 1, further comprising:

displaying a visual representation of an object in the first application window;

receiving a user input that moves the object from the first application window to the second application window; and

displaying the visual representation of the object in the second application window.

12. The method of claim 11, wherein the user input corresponds to a drag and drop gesture.

13. The method of claim 1, further comprising:

displaying a page-switching button on the first application window for scrolling content displayed in the first application window;

detecting a user interaction with the page-switching button, wherein the user interaction indicates a user request to scroll the content; and

displaying a page-switching indicator on the first application window in response to detecting the user interaction with the page-switching button.

14. The method of claim 1, further comprising:

providing a switch for user interface swap; and

in response to detecting a first user interaction with the switch, switching the desktop interface to a user interface other than the desktop interface, wherein the switching comprises displaying the user interface on the screen of the mobile device.

15. The method of claim 14, further comprising

displaying the desktop interface on the screen of the mobile device in response to detecting a second user interaction with the switch.

16. The method of claim 1, further comprising adjusting the size of the first application window in response to detecting a first user interaction with the adjustment button.

17. The method of claim 16, further comprising maximizing the first application window in response to detecting a second user interaction with the adjustment button.

18. A mobile device comprising:

a display; and

a processor communicatively coupled to the display, the processor to:

cause a desktop interface to be displayed on the display; and

cause a first application window and a second application window to be displayed on the desktop interface, wherein the first application window corresponds to a first application running on the mobile device, wherein the second application window corresponds to a second application running on the mobile device, wherein at least a portion of the first application window does not overlap with the second application window, and

wherein the first application window comprises an adjustment button for adjusting the size of the first application window and maximizing the first application window.

19. The mobile device of claim 18, wherein the processor is further to cause a task bar to be presented on the desktop interface, wherein the taskbar comprises a first visual representation of the first application and a second visual representation of the second application.

20. The mobile device of claim 19, wherein the processor is further to cause a start menu button to be presented on the taskbar for displaying a start menu.

21. The mobile device of claim 18, wherein the mobile device further comprises a memory configured to store an application configuration list.

22. The mobile device of claim 21, wherein the application configuration list comprises:

a first size and a first coordinate corresponding to the first application; and

a second size and a second coordinate corresponding to the second application.

23. The mobile device of claim 22, wherein the processor is further to query the list for the first size and the first coordinate and to query the list for the second size and the second coordinate of the second application.

24. The mobile device of claim 23, wherein the processor is further to:

cause the first application window to be displayed based on the first size and the first

coordinate.; and

cause the second application window to be displayed based on the second size and the second coordinate.

25. The mobile device of claim 24, wherein the second size corresponds to a type of the second application.

26. The mobile device of claim 25, wherein the type of the second application is one of a phone application and a tablet application.

27. The mobile device of claim 18, wherein the processor is further to:

detect a user interaction with at least one of the first application window and the second application window; and

rearrange the first application window and the second application window based on the user interaction, wherein the rearranged first application window is partially overlapped with the second application window.

28. The mobile device of claim 18, wherein the first application window further comprises an adjustment button for adjusting the size of the first application window.

29. The mobile device of claim 18, wherein the processor is further to:

cause a page-switching button to be displayed on the first application window for scrolling content displayed in the first application window;

detect a user interaction with the page-switching button, wherein the user interaction indicates a user request to scroll the content; and

cause a page-switching indicator to be displayed on the first application window in response to detecting the user interaction with the page-switching button.

30. The mobile device of claim 18, wherein the processor is further to provide a switch for switching the desktop interface to a user interface other than the desktop interface.

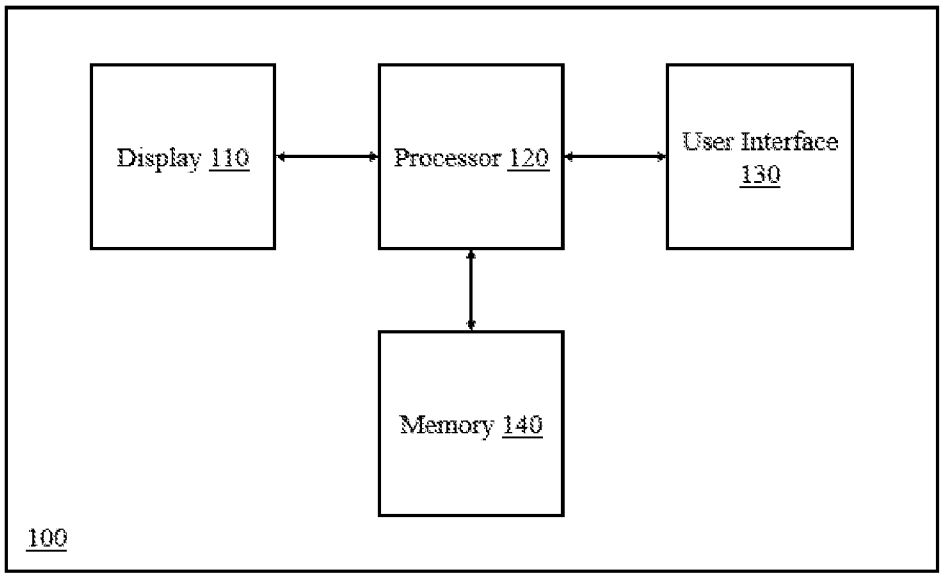
Display 110
Processor 120
User Interface 130
Memory 140
100

FIG. 1

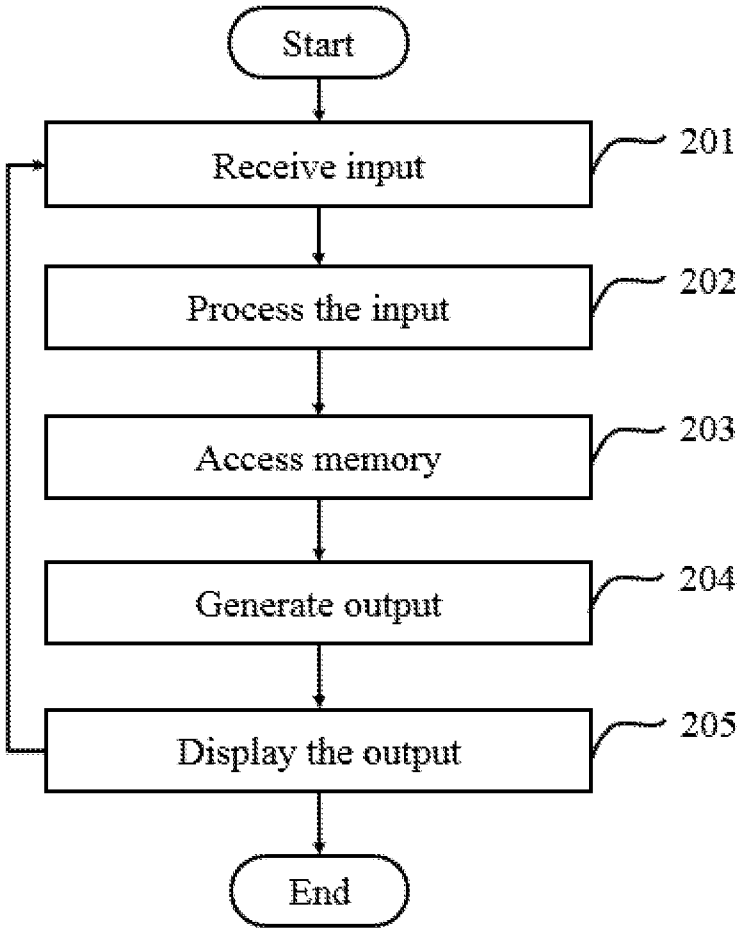
Start
Receive input
201
Process the input
202
Access memory
203
Generate output
204
Display the output
205
End

FIG. 2

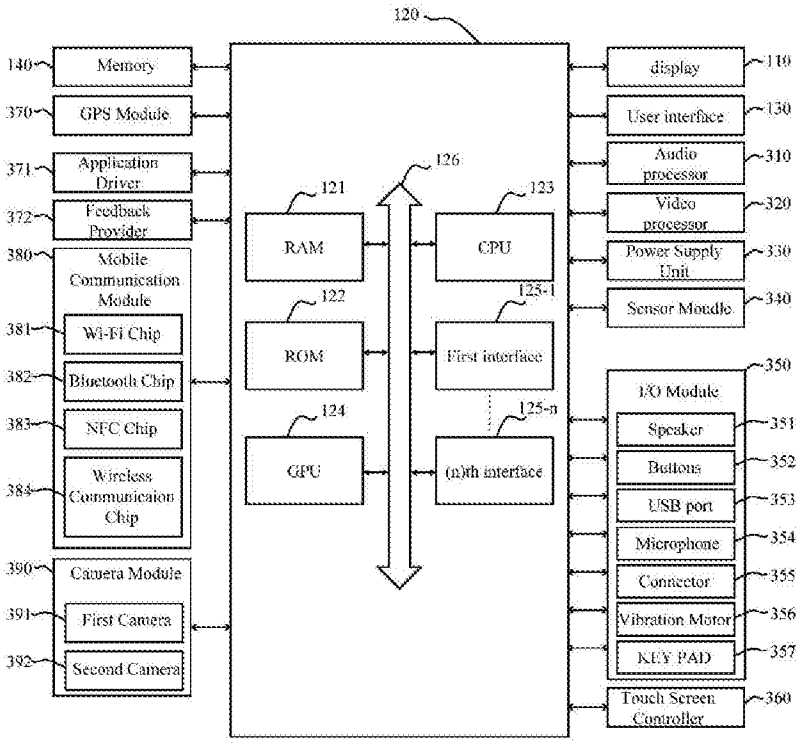
120
140
Memory
370
GPS Module
371
Application Driver
372
Feedback Provider
380
Mobile Communication Module
381
Wi-Fi Chip
382
Bluetooth Chip
383
NFC Chip
384
Wireless Communication Chip
390
Camera Module
391
First Camera
392
Second Camera
121
RAM
122
ROM
124
GPU
126
123
CPU
125-1
First interface
125-n
(n)th interface
display
110
User interface
130
Audio processor
310
Video processor
320
Power Supply Unit
330
Sensor Module
340
I/O Module
350
Speaker
351
Buttons
352
USB port
353
Microphone
354
Connector
355
Vibration Motor
356
KEY PAD
357
Touch Screen Controller
360

FIG. 3

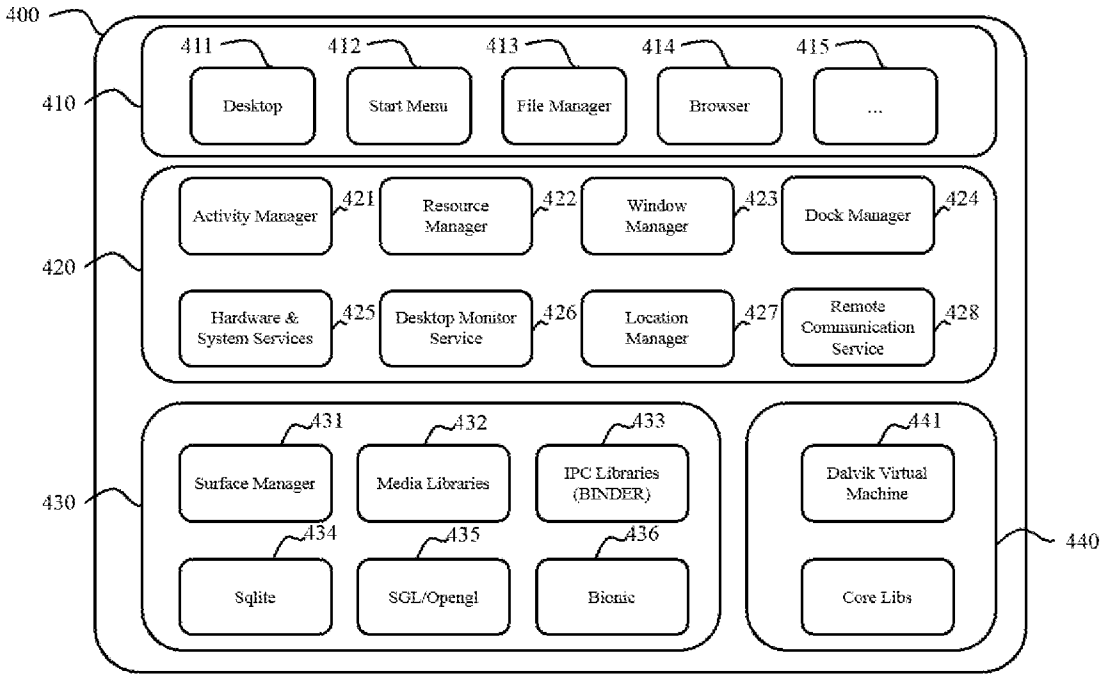
400
410
411
Desktop
412
Start Menu
413
File Manager
414
Browser
415
...
420
Activity Manager
421
Resource Manager
422
Window Manager
423
Dock Manager
424
Hardware & System Services
425
Desktop Monitor Service
426
Location Manager
427
Remote Communication Service
428
430
431
Surface Manager
432
Media Libraries
433
IPC Libraries (BINDER)
434
Sqlite
435
SGL/Opengl
436
Bionic
440
441
Dalvik Virtual Machine
Core Libs

FIG. 4

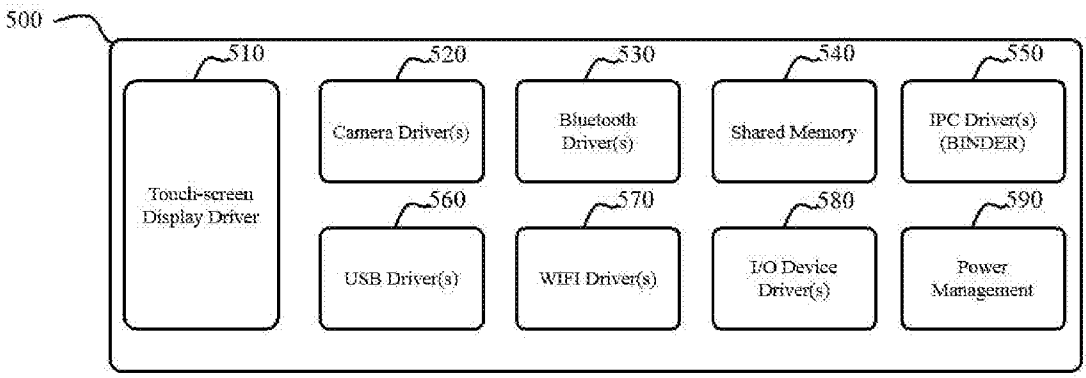
500
510
Touch-screen Display Driver
520
Camera Driver(s)
530
Bluetooth Driver(s)
540
Shared Memory
550
IPC Driver(s) (BINDER)
560
USB Driver(s)
570
WIFI Driver(s)
580
I/O Device Driver(s)
590
Power Management

FIG. 5

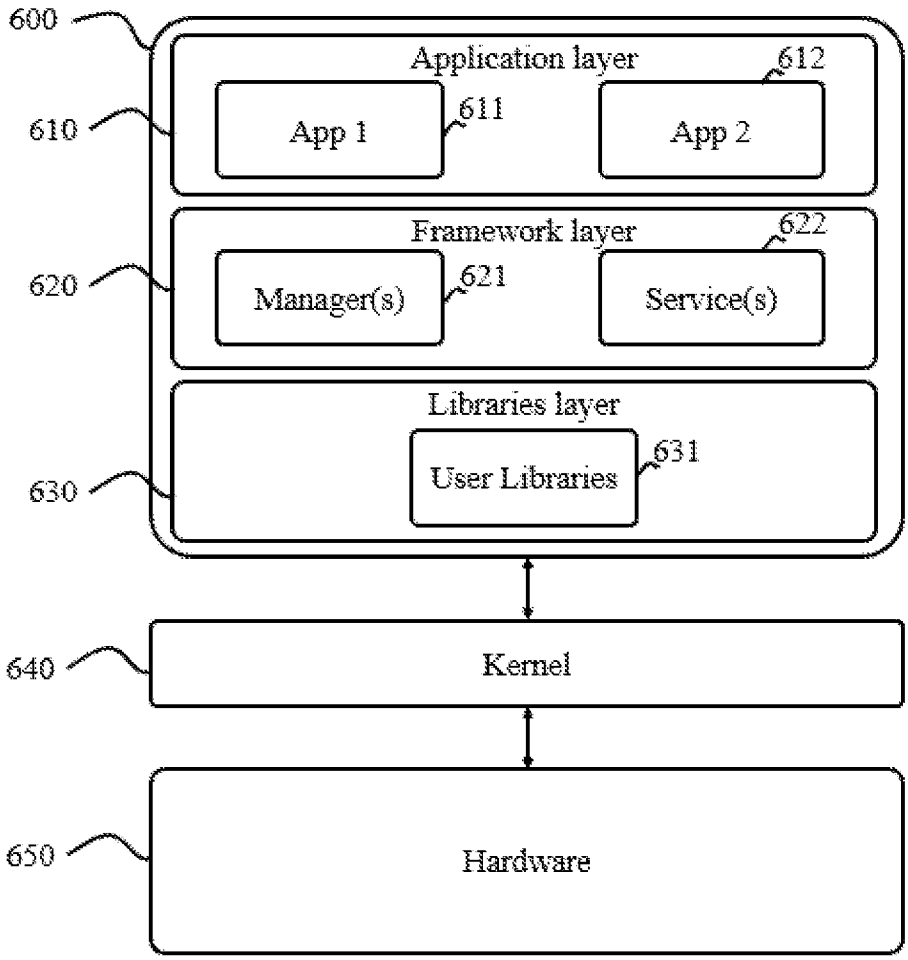
600
610
Application layer
612
App 1
611
App 2
620
Framework layer
622
Manager(s)
621
Service(s)
630
Libraries layer
User Libraries
631
640
Kernel
650
Hardware

FIG. 6

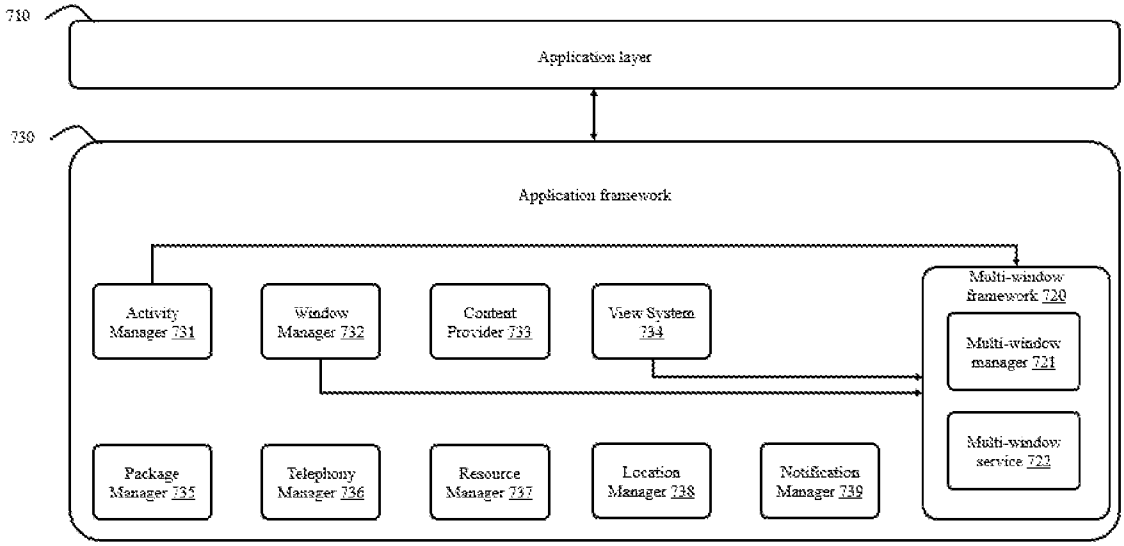
710
Application layer
730
Application framework
Activity Manager 731
Window Manager 732
Content Provider 733
View System 734
Multi-window framework 720
Multi-window manager 721
Multi-window service 722
Package Manager 735
Telephony Manager 736
Resource Manager 737
Location Manager 738
Notification Manager 739

FIG. 7

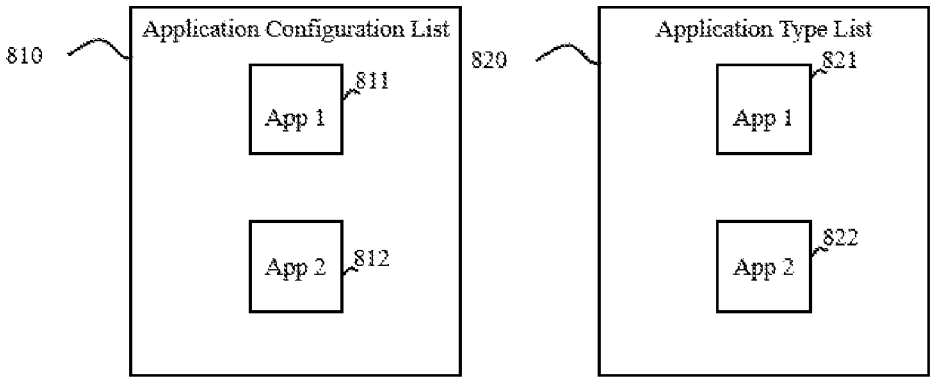
810
Application Configuration List
App 1
811
App 2
812
820
Application Type List
App 1
821
App 2
822

FIG. 8

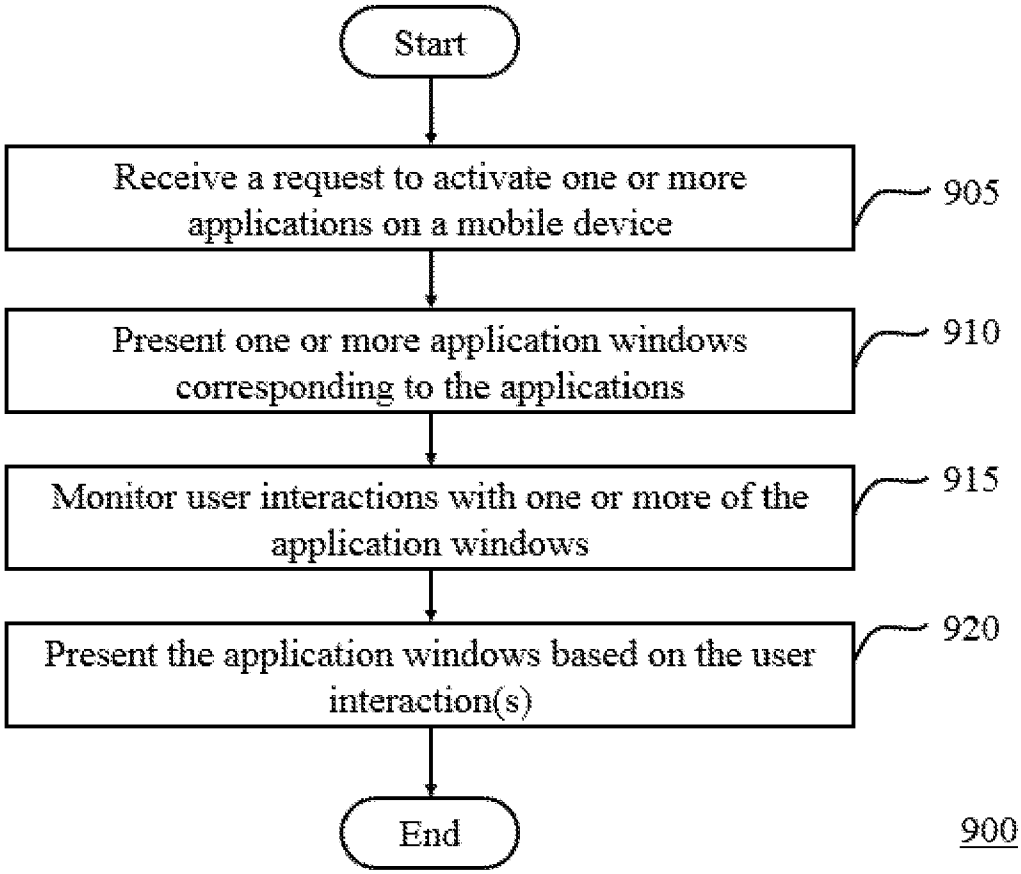
Start
Receive a request to activate one or more applications on a mobile device
905
Present one or more application windows corresponding to the applications
910
Monitor user interactions with one or more of the application windows
915
Present the application windows based on the user interaction(s)
920
End
900

FIG. 9A

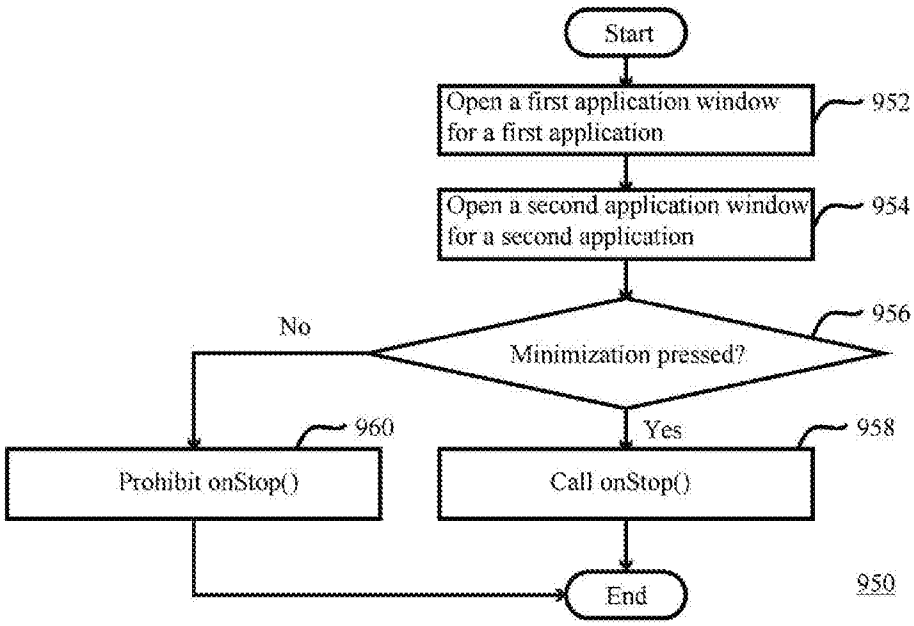
Start
Open a first application window for a first application
952
Open a second application window for a second application
954
Minimization pressed?
956
No
Yes
Prohibit onStop()
960
Call onStop()
958
End
950

FIG. 9B

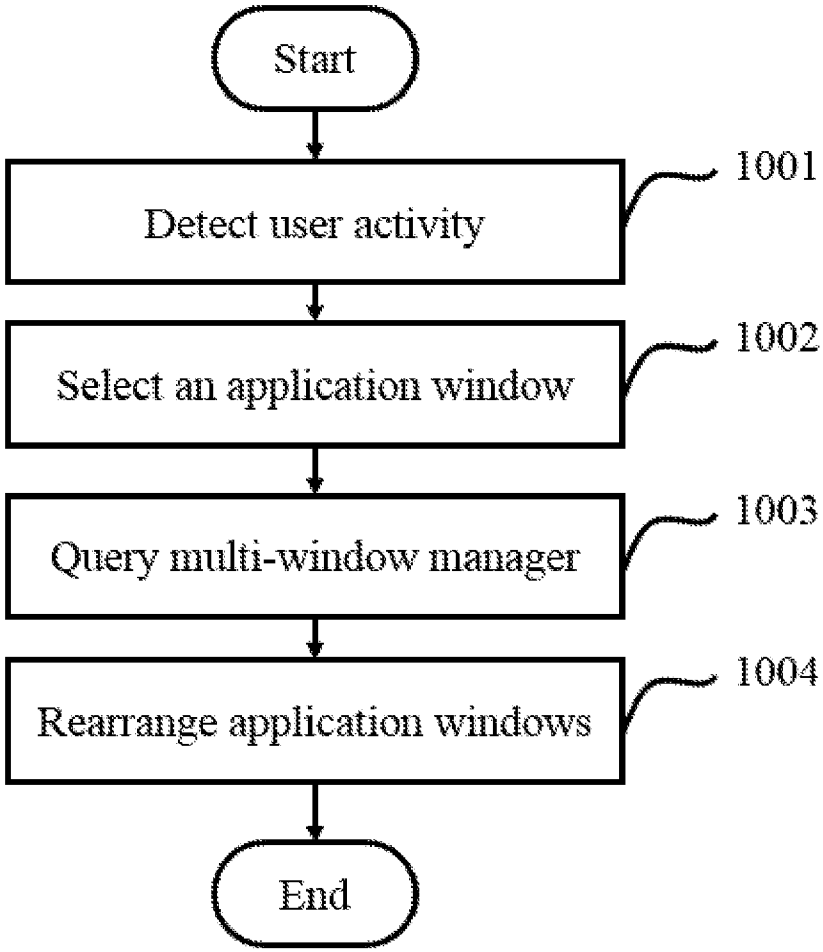
Start
Detect user activity
1001
Select an application window
1002
Query multi-window manager
1003
Rearrange application windows
1004
End

FIG. 10

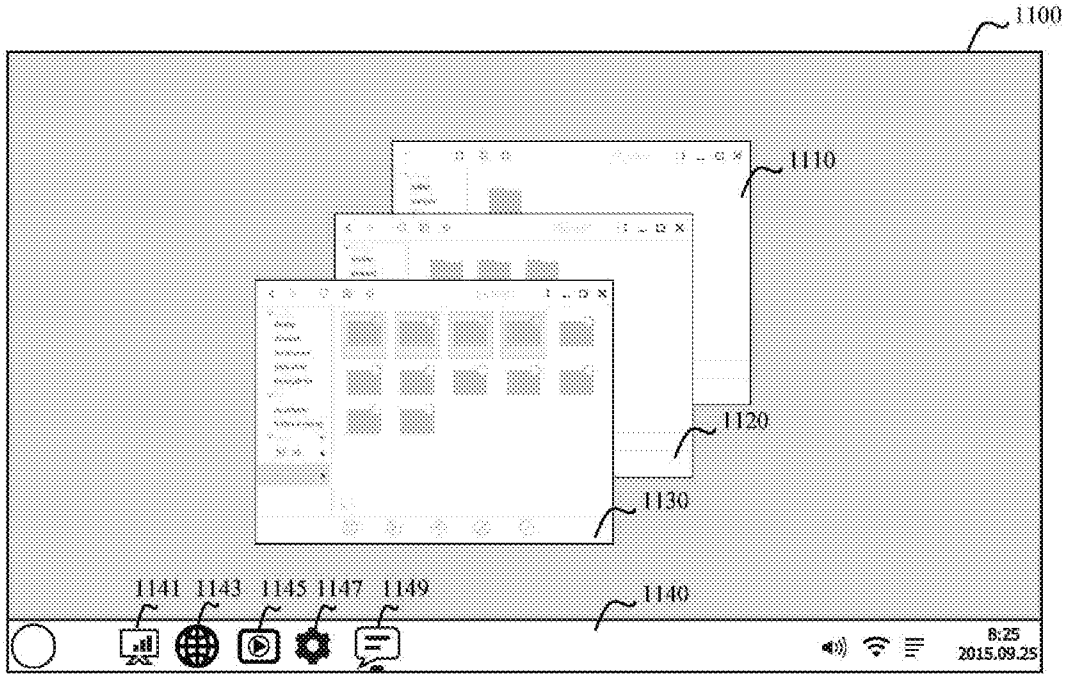
1100
1110
1120
1130
1140
1141
1143
1145
1147
1149
8:25
2015.09.25

FIG. 11

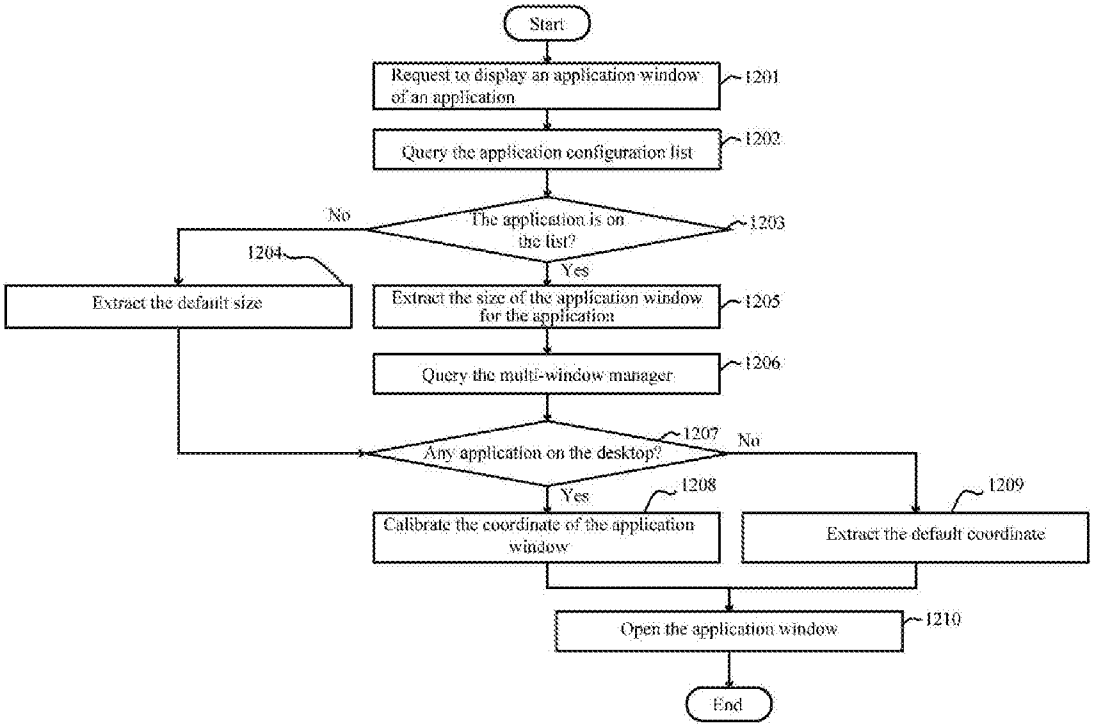
Start
Request to display an application window of an application
1201
Query the application configuration list
1202
The application is on the list?
1203
No
Yes
1204
Extract the default size
Extract the size of the application window for the application
1205
Query the multi-window manager
1206
Any application on the desktop?
1207
No
Yes
1208
Calibrate the coordinate of the application window
1209
Extract the default coordinate
Open the application window
1210
End

FIG. 12

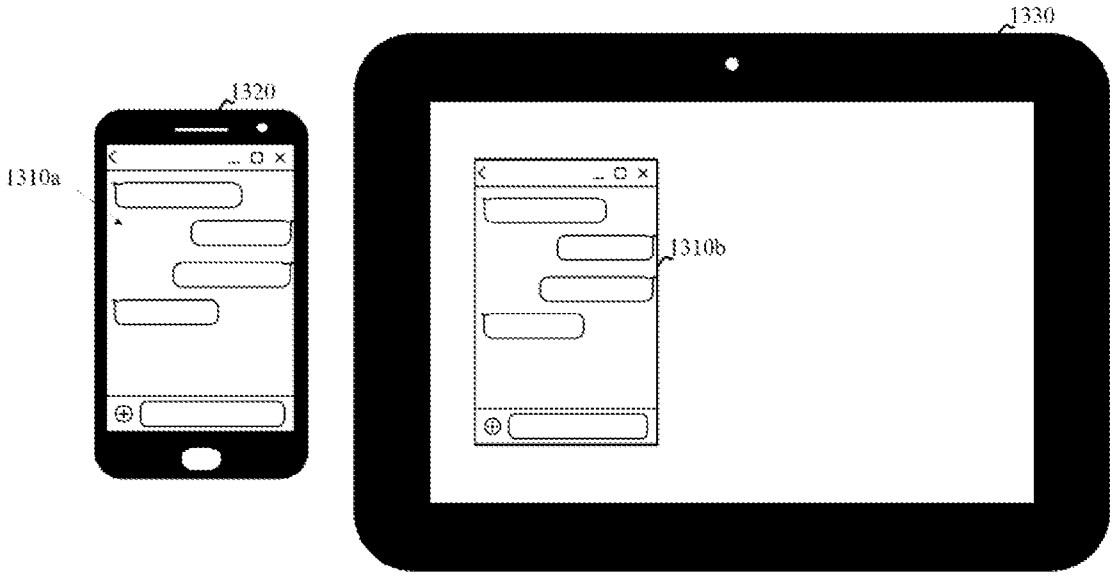

FIG. 13

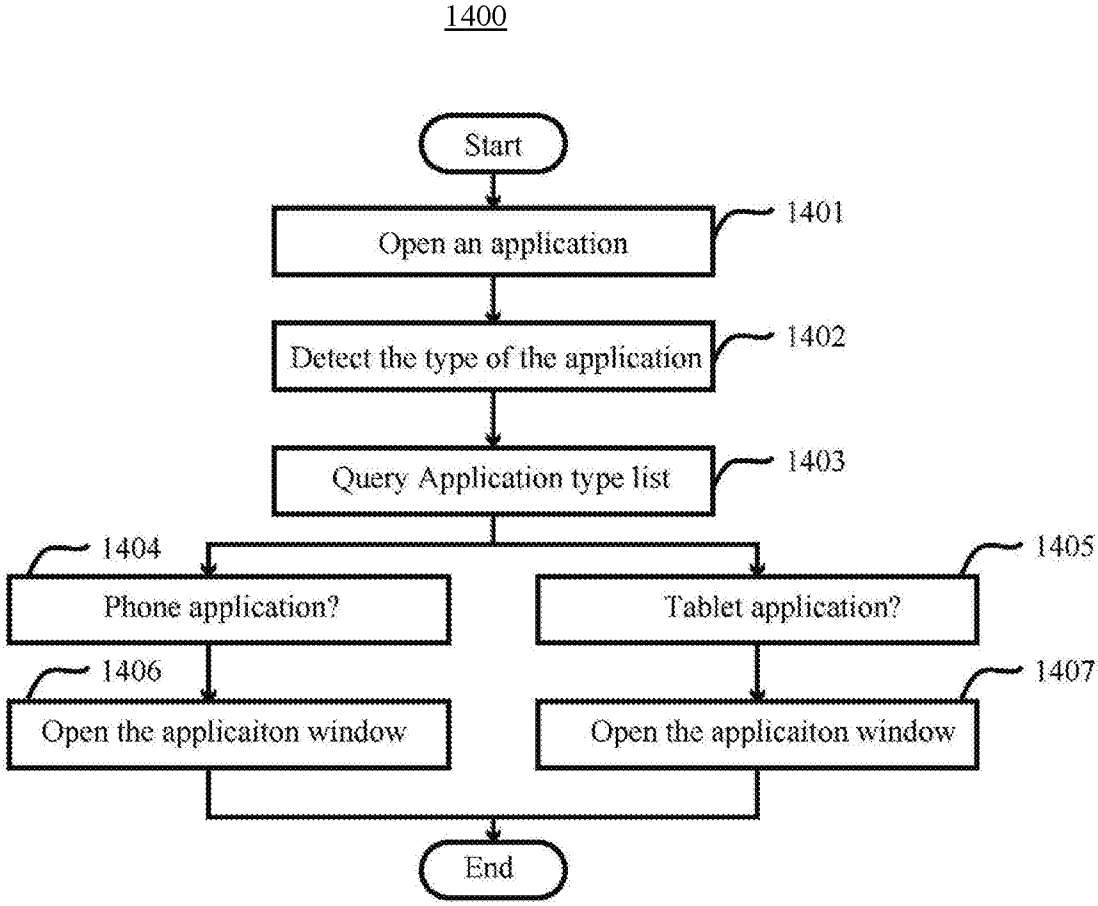
1400
Start
Open an application
1401
Detect the type of the application
1402
Query Application type list
1403
1404
Phone application?
1405
Tablet application?
1406
Open the applicaiton window
1407
Open the applicaiton window
End

FIG. 14

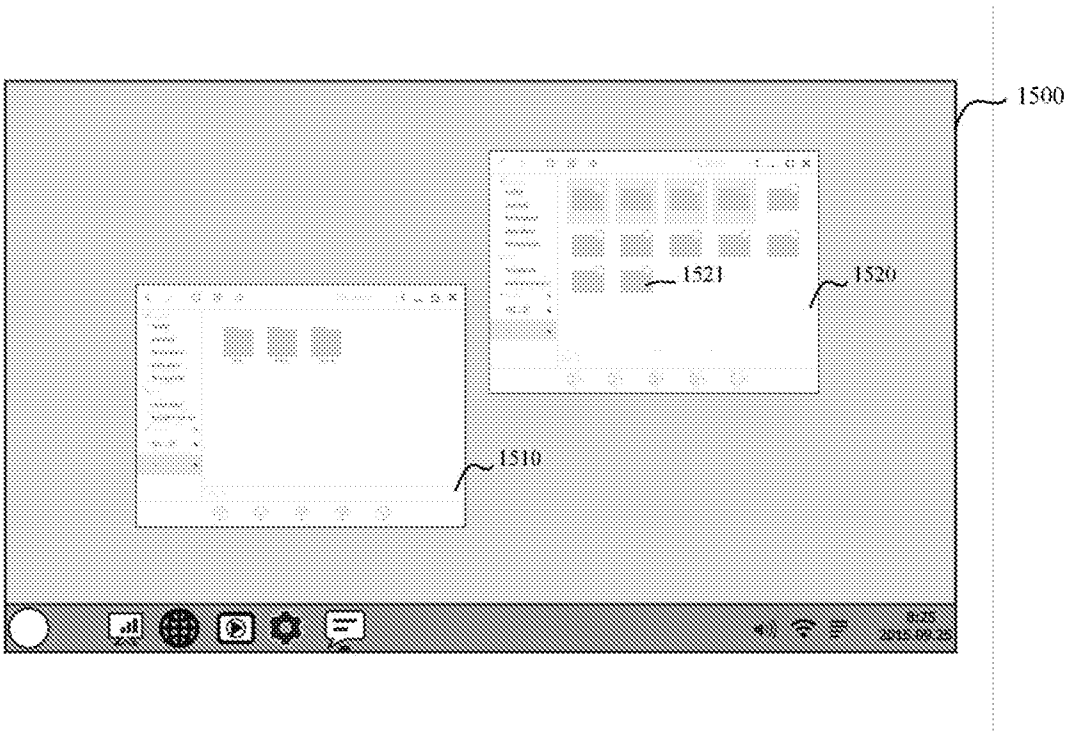

FIG. 15

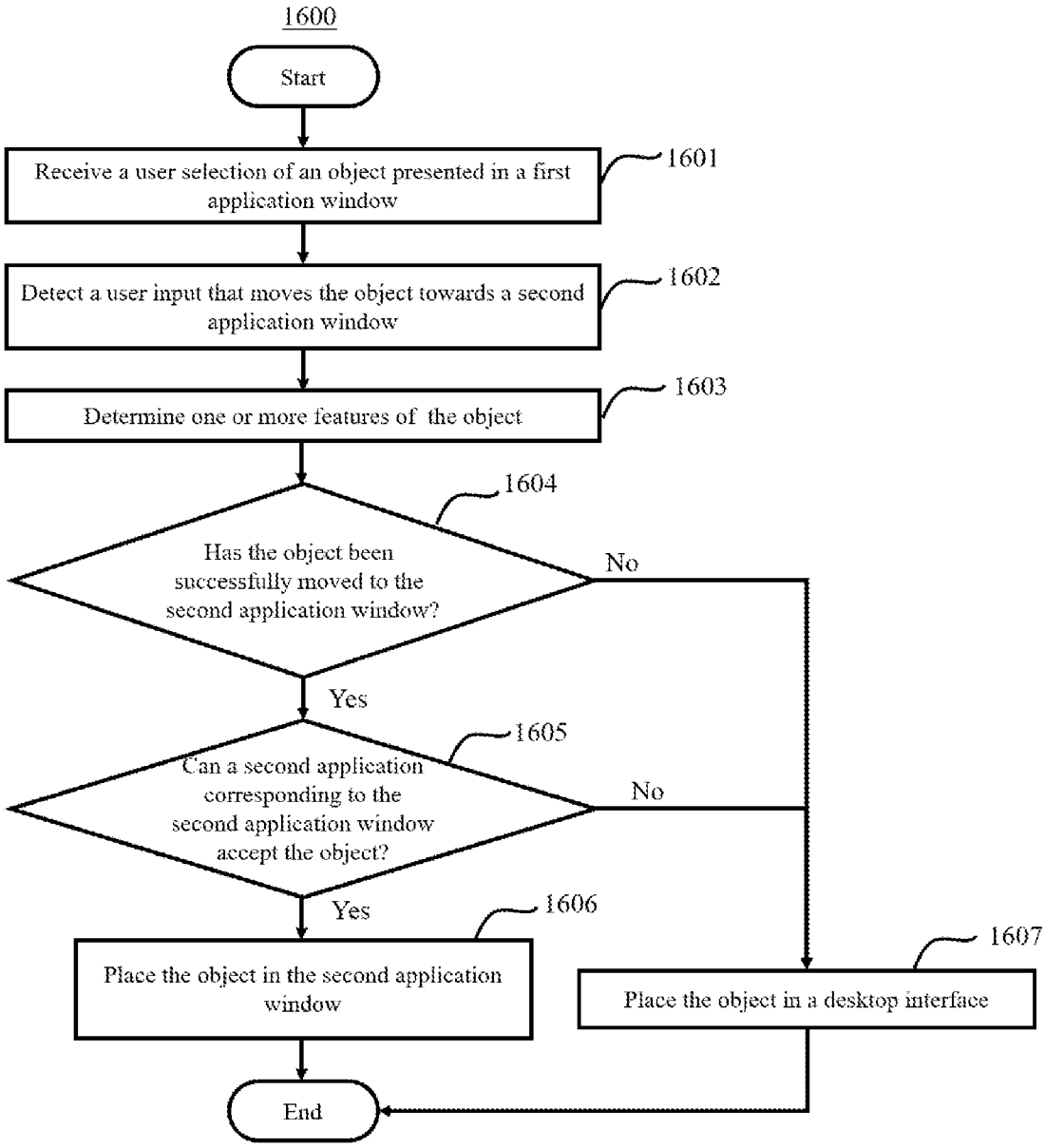
1600
Start
Receive a user selection of an object presented in a first application window
1601
Detect a user input that moves the object towards a second application window
1602
Determine one or more features of the object
1603
1604
Has the object been successfully moved to the second application window?
No
Yes
1605
Can a second application corresponding to the second application window accept the object?
No
Yes
1606
Place the object in the second application window
1607
Place the object in a desktop interface
End

FIG. 16

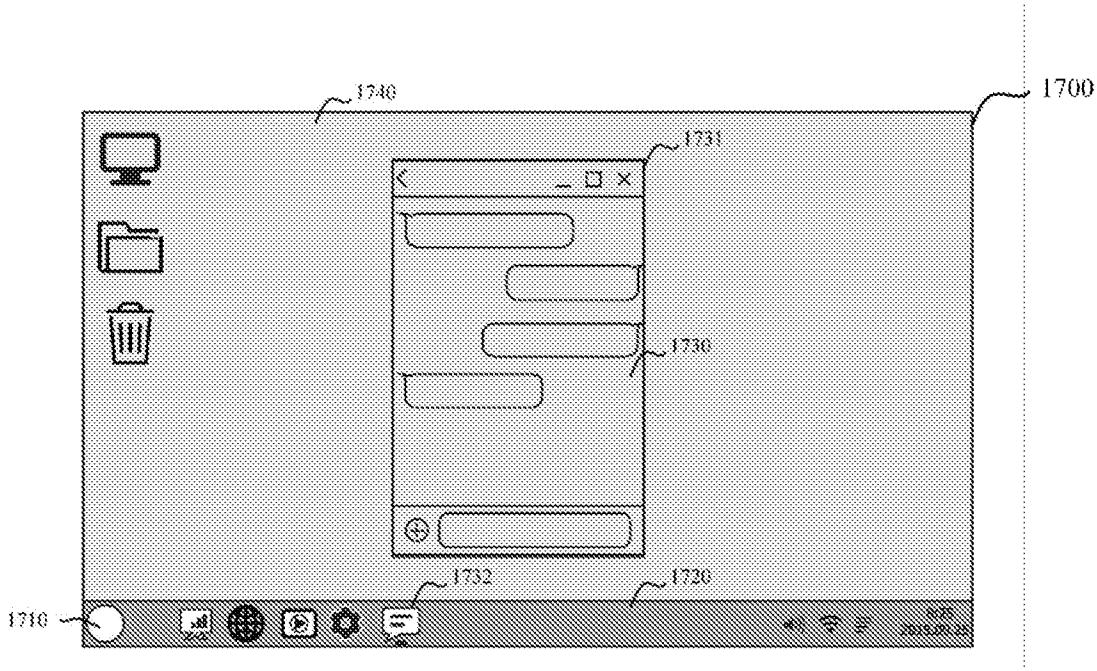

FIG. 17

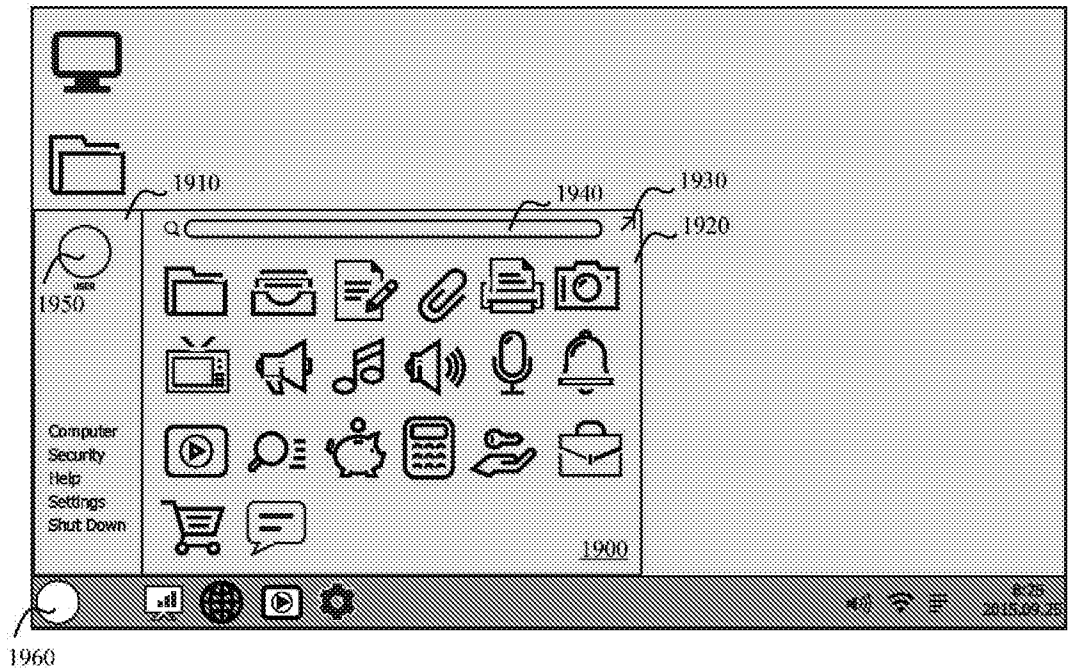
1910
1940
1930
1920
1950
Computer
Security
Help
Settings
Shut Down
1900
1960

FIG. 18

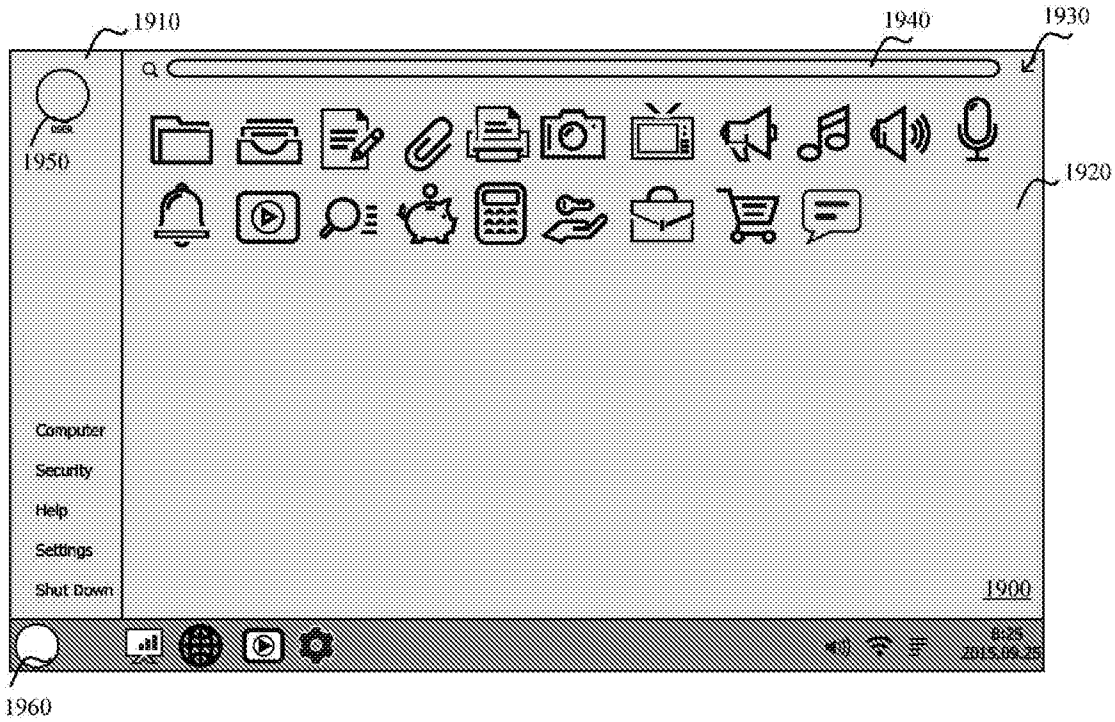
1910
1940
1930
1950
1920
Computer
Security
Help
Settings
Shut Down
1900
1960

FIG. 19

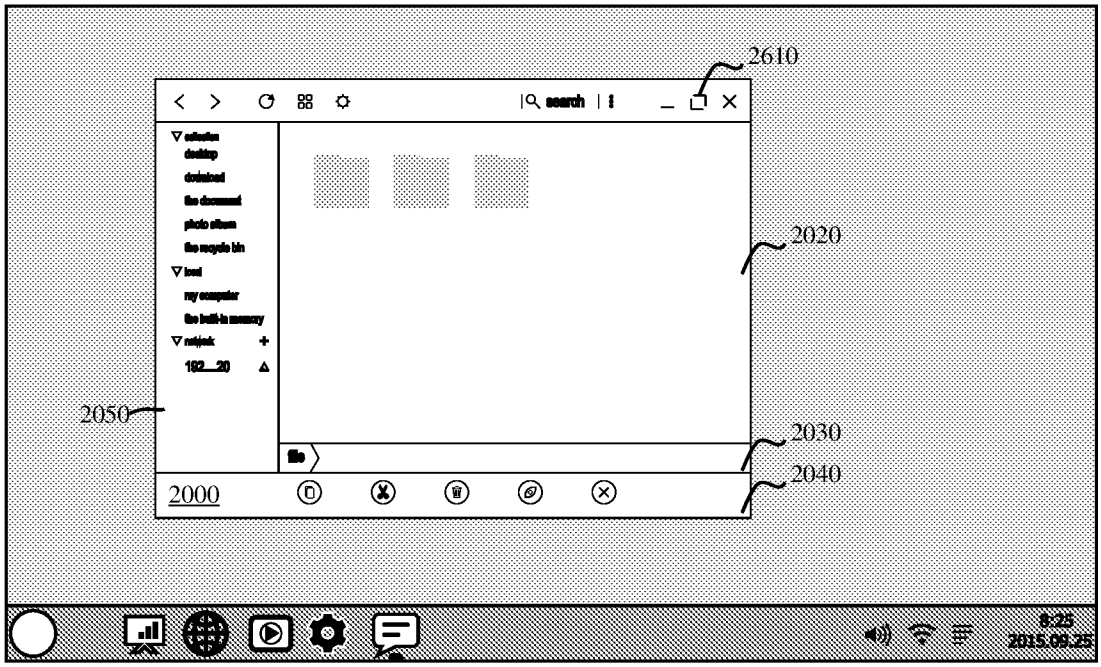

FIG. 20

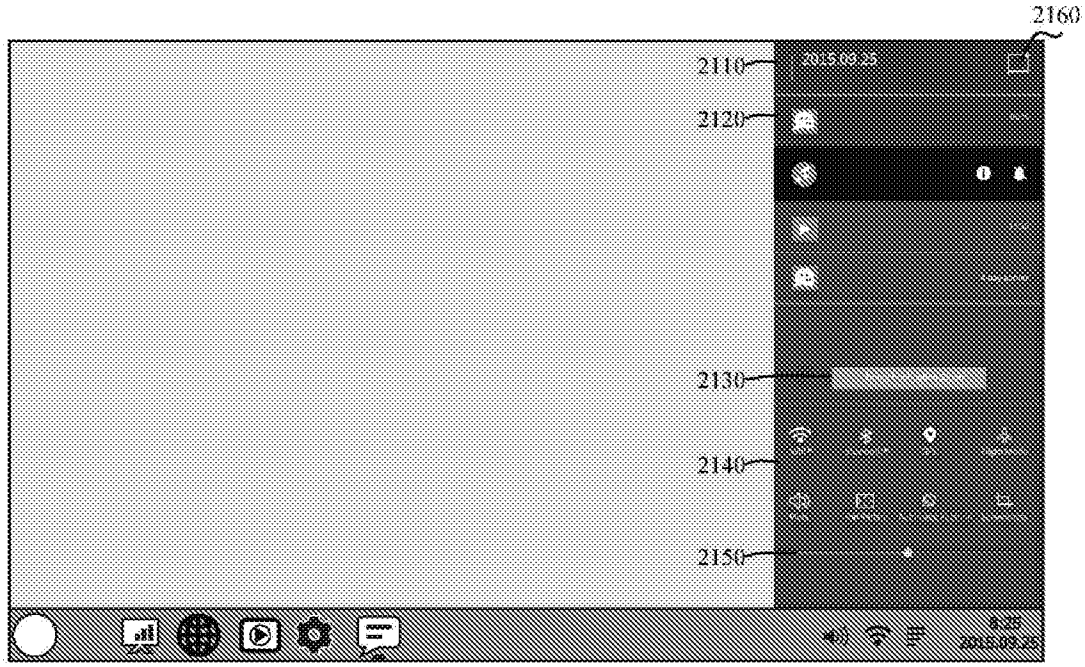

FIG. 21

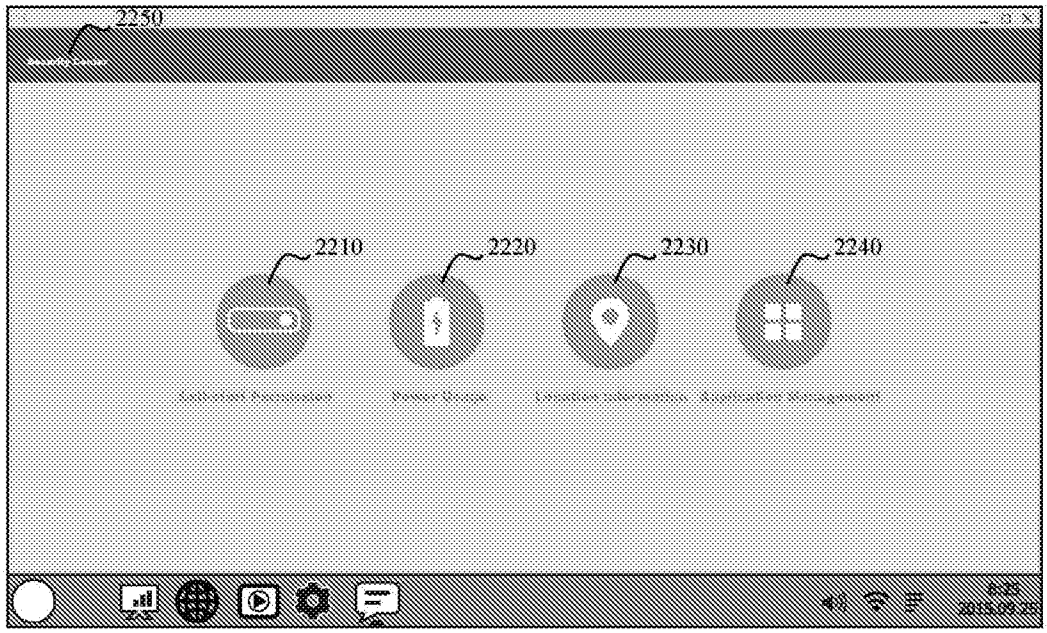

FIG. 22

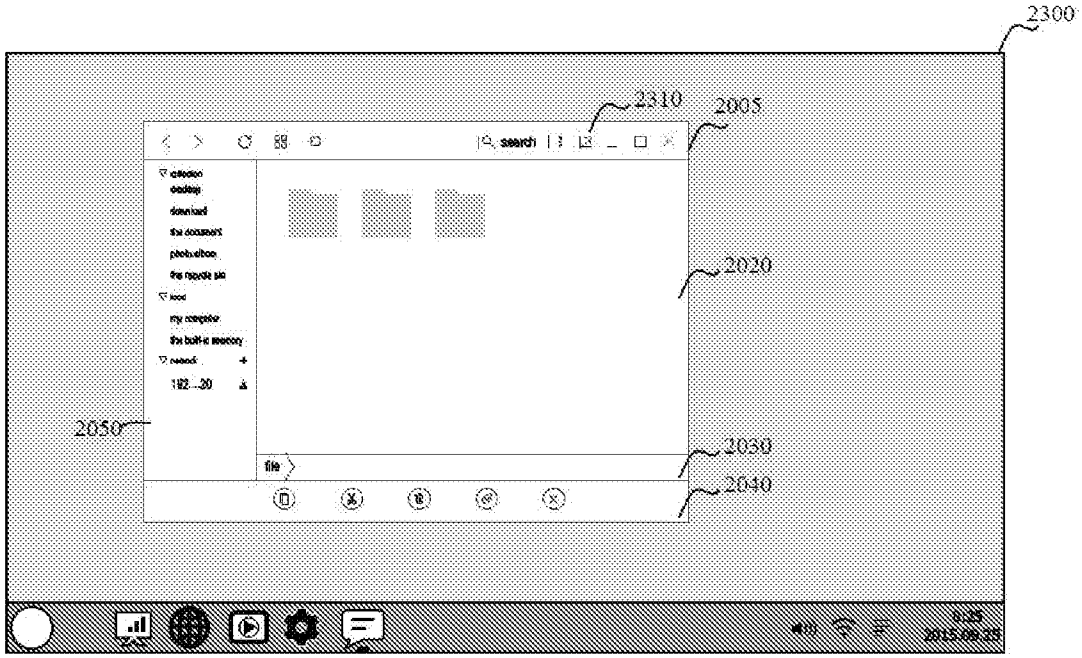

FIG. 23

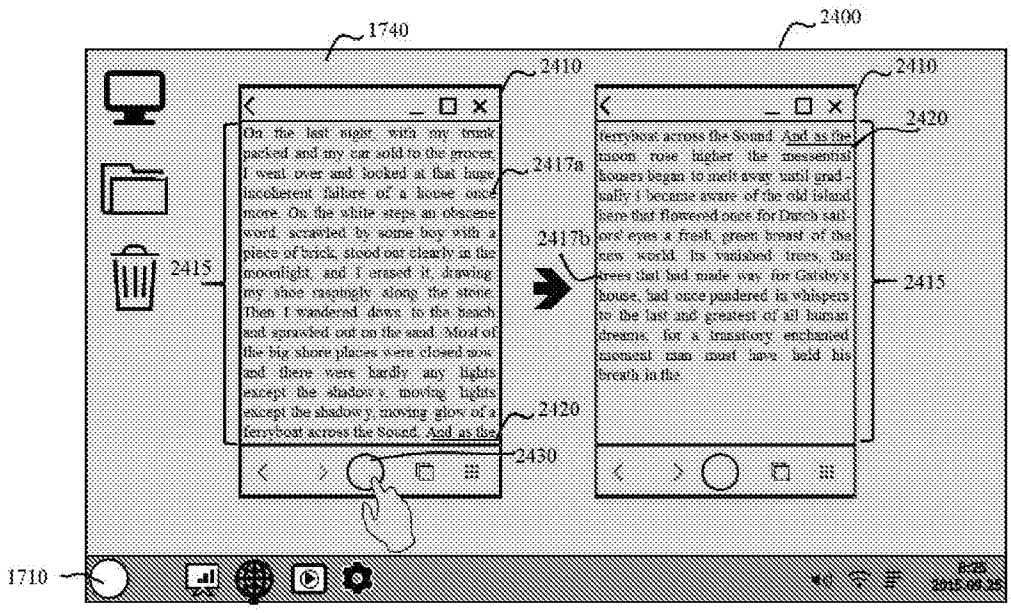
1740
2400
2410
2410
2420
2417a
2417b
2415
2415
2420
2430
1710
On the last night with my trunk packed and my car sold to the grocer, I went over and looked at that huge incoherent failure of a house once more. On the white steps an obscene word, scrawled by some boy with a piece of brick, stood out clearly in the moonlight, and I erased it, drawing my shoe raspingly along the stone. Then I wandered down to the beach and sprawled out on the sand. Most of the big shore places were closed now and there were hardly any lights except the shadowy, moving lights except the shadowy, moving glow of a ferryboat across the Sound. And as the
ferryboat across the Sound. And as the moon rose higher the mesential houses began to melt away until gradually I became aware of the old island here that flowered once for Dutch sailors' eyes a fresh, green breast of the new world. Its vanished trees, the trees that had made way for Gatsby's house, had once pandered in whispers to the last and greatest of all human dreams; for a transitory enchanted moment man must have held his breath in the

FIG. 24

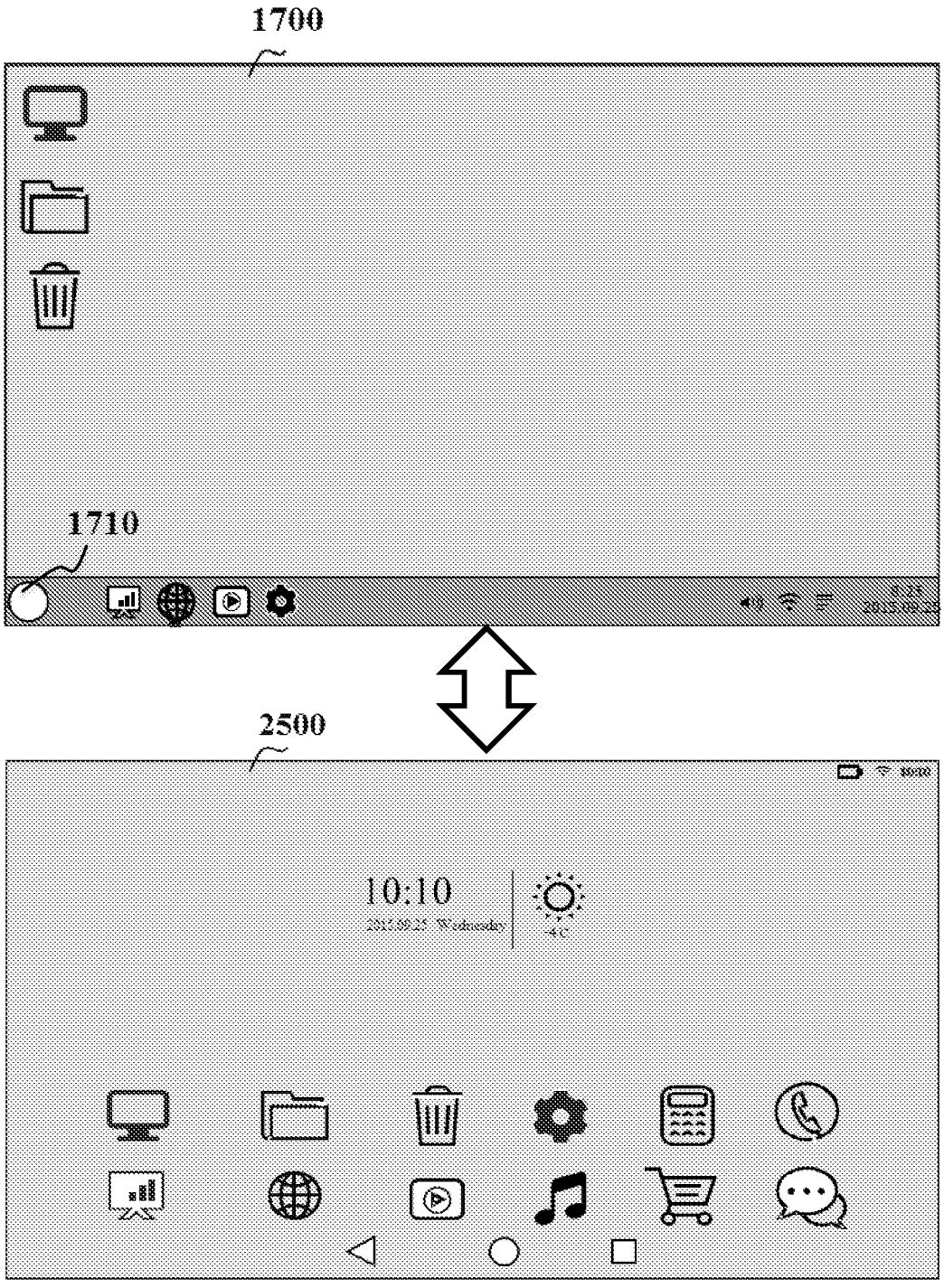

FIG. 25

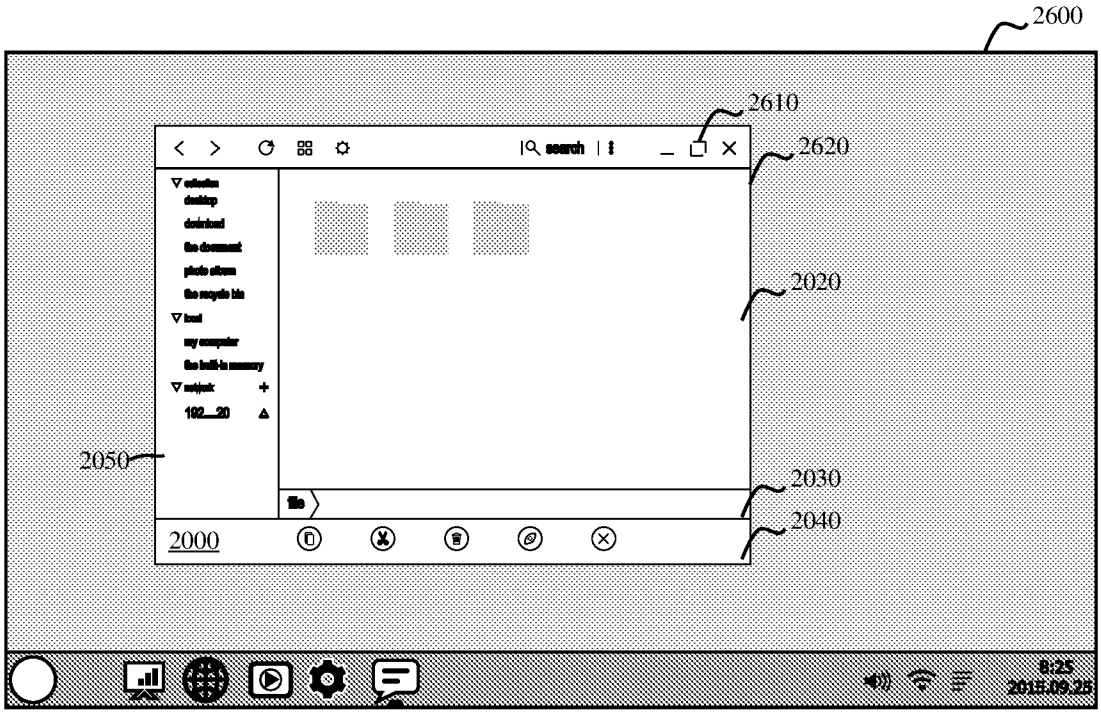

FIG. 26

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2016/085678

A. CLASSIFICATION OF SUBJECT MATTER

G06F 3/0488(2013.01)i

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)

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)

CNKI,CNPAT,WPI,EPODOC:mobile,operating,system,application,window,overlap,adjustment,maximize,button, configuration,list

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 103425531 A (INSYDE SOFTWARE CORP.) 04 December 2013 (2013-12-04) description, paragraphs [0036]-[0098], figures 3, 11 and 16	1-30
X	CN 103853451 A (SAMSUNG ELECTRONICS CO., LTD.) 11 June 2014 (2014-06-11) description, paragraphs [0152]-[0156], figures 12A-12E	1-30
A	CN 1811686 A (WEI, XINCHENG) 02 August 2006 (2006-08-02) the whole document	1-30
A	CN 103019609 A (GUANG DONG OPPO MOBILE TELECOMMUNICATIONS CORP., LTD) 03 April 2013 (2013-04-03) the whole document	1-30
A	CN 104750351 A (BYD CO., LTD.) 01 July 2015 (2015-07-01) the whole document	1-30
A	US 2011293083 A1 (LARSON, CURT T.) 01 December 2011 (2011-12-01) the whole document	1-30

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 August 2016	**23 September 2016**

Name and mailing address of the ISA/CN	Authorized officer
STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088** **China**	**SHI,Wenqing**
Facsimile No. **(86-10)62019451**	Telephone No. **(86-10)010-62413411**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2016/085678

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	103425531	A	04 December 2013	US	2014365933	A1	11 December 2014
				TW	201447726	A	16 December 2014
CN	103853451	A	11 June 2014	WO	2014088350	A1	12 June 2014
				EP	2741199	A1	11 June 2014
				US	2014164941	A1	12 June 2014
				CN	103853427	A	11 June 2014
				EP	2741192	A2	11 June 2014
				EP	2741190	A2	11 June 2014
				WO	2014088348	A1	12 June 2014
				WO	2014088342	A1	12 June 2014
				US	2014164966	A1	12 June 2014
				US	2014164957	A1	12 June 2014
				KR	20140073396	A	16 June 2014
				KR	20140073373	A	16 June 2014
				KR	20140073372	A	16 June 2014
				KR	20140073371	A	16 June 2014
				KR	20140074141	A	17 June 2014
				JP	2014116010	A	26 June 2014
				AU	2013355443	A1	02 July 2015
				CN	104903830	A	09 September 2015
CN	1811686	A	02 August 2006		None		
CN	103019609	A	03 April 2013	CN	103019609	B	03 February 2016
CN	104750351	A	01 July 2015		None		
US	2011293083	A1	01 December 2011	US	8638913	B2	28 January 2014