



(51) International Patent Classification:
G06F 9/451 (2018.01)

(21) International Application Number:
PCT/CN2018/125485

(22) International Filing Date:
29 December 2018 (29.12.2018)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **BEIJING DIDI INFINITY TECHNOLOGY AND DEVELOPMENT CO., LTD.** [CN/CN]; Building 34, No. 8 Dongbeiwang West Road, Haidian District, Beijing 100193 (CN).

(72) Inventor: **YANG, Kai**; Building 34, No. 8 Dongbeiwang West Road, Haidian District, Beijing 100193 (CN).

(74) Agent: **METIS IP (CHENGDU) LLC**; Room 808, 8th Floor, Building B7, Block D, Tianfu Jingrong Center, No. 99 West Section Of Hupan Road, Tianfu New District, Chengdu, Sichuan 610213 (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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,

HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: SYSTEM AND METHOD FOR INFORMATION DISPLAY

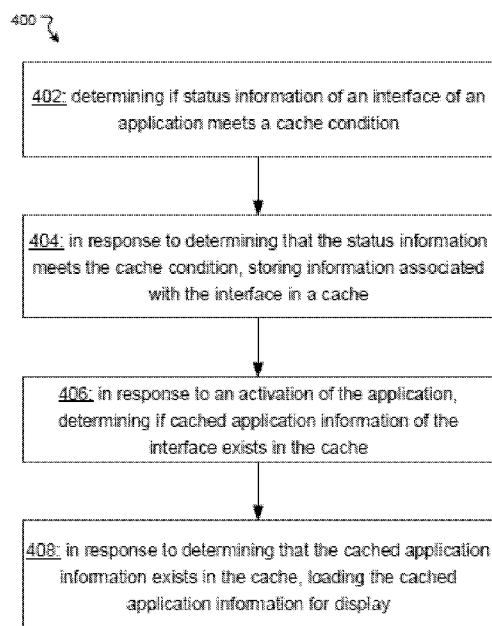


FIG. 4

(57) Abstract: A computer-implemented method for information display includes: determining if status information of an interface of an application meets a cache condition; in response to determining that the status information meets the cache condition, storing information associated with the interface in a cache; in response to an activation of the application, determining if cached application information of the interface exists in the cache; and in response to determining that the cached application information exists in the cache, loading the cached application information for display.

SYSTEM AND METHOD FOR INFORMATION DISPLAY

FIELD OF THE INVENTION

[0001] This disclosure generally relates to computer technology, and in particular, to systems
5 and methods for information display.

BACKGROUND

[0002] Applications on computing devices such as mobile phones and computers are popular for performing various tasks. Sometimes, the application operation may be disrupted
10 for various reasons such as poor internet connection or insufficient storage, and it may take a long time for it to restore. Such disruption may prevent the users from viewing important information last displayed and even cause failure to the task performance.

SUMMARY

[0003] Various embodiments of the present disclosure include systems, methods, and non-transitory computer-readable storage media for information display.

[0004] According to one aspect, a computer-implemented method for information display comprises: determining if status information of an interface of an application meets a cache condition; in response to determining that the status information meets the cache condition,
20 storing information associated with the interface in a cache; in response to an activation of the application, determining if cached application information of the interface exists in the cache; and in response to determining that the cached application information exists in the cache, loading the cached application information for display.

[0005] In some embodiments, the activation of the application comprises: a restart of the
25 application.

[0006] In some embodiments, in response to the activation of the application, determining if the cached application information of the interface exists in the cache comprises: in response to the activation of the application, determining if the application is killed; and in response to determining that the application is killed, determining if the cached application
30 information of the interface exists in the cache.

[0007] In some embodiments, the information associated with the interface comprises at least one of: at least a portion of currently displayed information on the interface or at least a portion of background information associated with the currently displayed information.

[0008] In some embodiments, the method further comprises: requesting for a latest status of the interface from a server; in response to obtaining the latest status from the server, restoring the interface in the latest status for display; and in response to failing to obtain the current status from the server, displaying a current interface of the application and triggering error notification.

[0009] In some embodiments, before determining if the cached application information of the interface exists in the cache, the method further comprises: displaying a current interface of the application.

[0010] In some embodiments, the method further comprises: in response to determining that the cached application information of the interface does not exist, loading a designated interface of the application for display. In one example, the designated interface comprises a home page of the application. In another example, storing the information associated with the interface in the cache comprises storing currently displayed information associated with a timestamp in the cache; and the designated interface comprises a cached interface with a latest timestamp.

[0011] In some embodiments, the information associated with the interface is stored as a page structure.

[0012] In some embodiments, the cache condition comprises the interface being an “order placed” or “boarding vehicle” interface, for both of which a ride hailing order has been placed through the application by a user, and the user has not started the ride; the cached application information comprises at least one of: a map, order information, or command triggers; the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, or vehicle information for the ride hailing order; and the command triggers comprise at least one of: returning to a previous interface, initiating instant messenger communication with the driver, initiating phone call with the driver, cancelling the ride hailing order, sharing the rider hailing order, or initiating communication with customer service.

[0013] In some embodiments, the cache condition comprises the interface being an “in ride” interface, for which a ride hailing order has been placed through the application by a user, and the user is riding in a vehicle that picked up the user; the cached application information comprises at least one of: a map, order information, or command triggers; the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing

order, or vehicle information for the ride hailing order; and the command triggers comprise at least one of: returning to a previous interface, initiating instant messenger communication with the driver, initiating phone call with the driver, showing a cost for the ride hailing order, sharing the rider hailing order, or initiating communication with customer service.

5 **[0014]** In some embodiments, the cache condition comprises the interface being a “ride finished” interface, for which a ride hailing order was placed through the application by a user, and the user has finished the ride; the cached application information comprises at least one of: a map or order information; and the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map,
10 information of a driver for the ride hailing order, vehicle information for the ride hailing order, or details of cost for the rider hailing order.

[0015] According to another aspect, a system for information display comprises one or more processors and a non-transitory computer-readable storage medium coupled to the one or more processors, the storage medium storing instructions that, when executed by the one
15 or more processors, cause the system to perform operations comprising: determining if status information of an interface of an application meets a cache condition; in response to determining that the status information meets the cache condition, storing information associated with the interface in a cache; in response to an activation of the application, determining if cached application information of the interface exists in the cache; and in
20 response to determining that the cached application information exists in the cache, loading the cached application information for display.

[0016] According to another aspect, a non-transitory computer-readable storage medium is coupled to one or more processors, the storage medium storing instructions that, when executed by the one or more processors, cause the one or more processors to perform
25 operations comprising: determining if status information of an interface of an application meets a cache condition; in response to determining that the status information meets the cache condition, storing information associated with the interface in a cache; in response to an activation of the application, determining if cached application information of the interface exists in the cache; and in response to determining that the cached application information
30 exists in the cache, loading the cached application information for display.

[0017] According to one aspect, a computer-implemented method for information display comprises: determining if status information of an interface of an application meets a cache condition; in response to determining that the status information meets the cache condition, storing information associated with the interface in a cache; in response to a disruption to the

application's operation, determining if cached application information of the interface exists in the cache; and in response to determining that the cached application information exists in the cache, loading the cached application information for display. In some embodiments, the disruption comprises: a disruption to the application's connectivity with internet. In some
5 embodiments, the disruption comprises: the application being killed.

[0018] According to another aspect, a system for information display comprises one or more processors and a non-transitory computer-readable storage medium coupled to the one or more processors, the storage medium storing instructions that, when executed by the one or more processors, cause the system to perform operations comprising: determining if status
10 information of an interface of an application meets a cache condition; in response to determining that the status information meets the cache condition, storing information associated with the interface in a cache; in response to a disruption to the application's operation, determining if cached application information of the interface exists in the cache; and in response to determining that the cached application information exists in the cache,
15 loading the cached application information for display.

[0019] According to another aspect, a non-transitory computer-readable storage medium is coupled to one or more processors, the storage medium storing instructions that, when executed by the one or more processors, cause the one or more processors to perform operations comprising: determining if status information of an interface of an application
20 meets a cache condition; in response to determining that the status information meets the cache condition, storing information associated with the interface in a cache; in response to a disruption to the application's operation, determining if cached application information of the interface exists in the cache; and in response to determining that the cached application information exists in the cache, loading the cached application information for display.

[0020] These and other features of the systems, methods, and non-transitory computer readable media disclosed herein, as well as the methods of operation and functions of the related elements of structure and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this
25 specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for purposes of illustration and description only and are not intended as a definition of the limits of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] Certain features of various embodiments of the present technology are set forth with particularity in the appended claims. A better understanding of the features and advantages of the technology will be obtained by reference to the following detailed description that sets forth illustrative embodiments, in which the principles of the invention are utilized, and the accompanying drawings of which:

[0022] FIG. 1 illustrates an exemplary system for information display, in accordance with various embodiments.

[0023] FIG. 2 illustrates a flowchart of an exemplary method for information display, in accordance with various embodiments.

[0024] FIG. 3A illustrates an exemplary interface for information display, in accordance with various embodiments.

[0025] FIG. 3B illustrates an exemplary interface for information display, in accordance with various embodiments.

[0026] FIG. 3C illustrates an exemplary interface for information display, in accordance with various embodiments.

[0027] FIG. 3D illustrates an exemplary interface for information display, in accordance with various embodiments.

[0028] FIG. 4 illustrates a flowchart of an exemplary method for information display, in accordance with various embodiments.

[0029] FIG. 5 illustrates an exemplary system for information display, in accordance with various embodiments.

[0030] FIG. 6 illustrates a block diagram of an exemplary computer system in which any of the embodiments described herein may be implemented.

DETAILED DESCRIPTION

[0031] Applications (APPs) on computing devices such as mobile phones and computers are popular among users for performing various tasks. For example, users install various applications to order food, purchase goods, hail transportation services, etc. While in use, some of the applications may display time-sensitive or otherwise important information, such as an on-going order, and the users may need constant and reliable access to such information for completing the order or service.

[0032] However, the application operation may be disrupted for various reasons. The disruption may be caused by the operating system of the computing device, by the user, by

the application itself, by another process, etc. The disruption may be relative to the application's normal function, for example, when an active application in use is closed out, is tapped out, lags in wireless connection, slows down in processing, malfunctions, etc. In one example, the computing device with the application running may experience insufficient
5 memory when cache memory is full. In this situation, the operating system may kill (or force-stop) the application, causing disruption. For Android and IOS systems, LowMemoryKiller or a similar protocol may kill the application. In another example, the application may have a lower priority than other applications and/or processes, such that the operating system may kill the application to release resources for those with higher priorities, causing disruption. In
10 another example, if the application needs to access internet (e.g., through wireless communication), disruption is caused when the connection to the internet is slow, interrupted, unresponsive, or otherwise not working. In another example, the user may force-close the application, causing disruption. In another example, the application may run into some internal bug, crash, and close out.

[0033] In current technologies, when the disruption occurs, it may take a long time for the application to restore the last displayed information or other information that the user may need. Such disruption may prevent the users from viewing important information last displayed and even cause failure to the task performance. For instance, if a user uses an online vehicle hailing application to order transportation, when waiting for the dispatched
20 vehicle, the user may need to check against the displayed information of vehicle to determine if an approaching vehicle is the coming to pick up the user. If the application operation is disrupted for poor wireless connection or another reason, the user may need to reset or restart the application to restore the interface that was last displaying the information of the vehicle. For existing methods, it often takes three seconds to first restore a home page of the
25 application, and then takes more than five seconds to restore the last displayed interface. That is, the user may have to wait for about eight seconds to restore the original information display, during which the ordered vehicle may be missed.

[0034] To at least mitigate the deficiencies of current technologies, provide a technical solution to the above-described technical problem, and improve computer functionalities, systems, methods, and non-transitory computer readable media for information display are disclosed. By using cache memory, the last displayed interface or other important information
30 can be timely restored after the disruption occurs. In some cases, it may just take up to about one second to restore the last displayed information.

[0035] FIG. 1 illustrates an exemplary system 100 for information display, in accordance with various embodiments. As shown in FIG. 1, the exemplary system 100 may include at least one computing system 102 that includes one or more processors 104 and one or more memories 106 coupled to the one or more processors 104. The memory 106 may be non-transitory and computer-readable. The system 102 may be implemented on or as various devices such as server, computer, etc. The system 102 above may be installed with appropriate software (e.g., vehicle hailing platform program, etc.) and/or hardware (e.g., wires, wireless connections, etc.) to access other devices of the system 100.

[0036] The system 100 may include one or more data stores (e.g., a data store 108) and one or more computing devices (e.g., a computing device 109) that are accessible to the system 102. In some embodiments, the system 102 may be configured to obtain data from the data store 108 (e.g., a database or dataset of user information) and/or the computing device 109 (e.g., a computer, a server, or a mobile phone). The system 102 may use the obtained data to provide operations of a vehicle hailing platform.

[0037] The system 100 may further include one or more computing devices (e.g., computing devices 110, 111, 112) coupled to the system 102. The computing devices may include devices such as mobile phone, tablet, computer, wearable device (smart watch), etc. The computing devices may transmit or receive data to or from the system 102.

[0038] An exemplary structure of the computing device is described with respect to computing device 110. The computing device 110 or a similar device may also be referred to as a system for information display. In various embodiments, the computing device 110 may include one or more processors 114 and one or more memories 116 coupled to the one or more processors 114. The memory 116 may be non-transitory and computer-readable. The memory 116 may store instructions that, when executed by the one or more processors 114, cause the computing device 110 (e.g., the one or more processors 114) to perform various operations described herein.

[0039] In some embodiments, the system 102 may implement an online information or service platform. The service may be associated with vehicles (e.g., cars, bikes, boats, airplanes, etc.), and the platform may be referred to as a vehicle hailing platform (or service hailing platform, ride order dispatching platform, etc.). The platform may accept requests for transportation, identify vehicles to fulfill the requests, arrange for pick-ups, and process transactions. For example, a user may use the computing device 110 (e.g., a mobile phone installed with a software application associated with the platform) to request transportation from the platform. The system 102 may receive the request and relay it to various vehicle

drivers (e.g., by posting the request to mobile phones carried by the drivers). A vehicle driver may use the computing device 111 (e.g., another mobile phone installed with the application associated with the platform) to accept the posted transportation request and obtain pick-up location information. Fees (e.g., transportation fees) can be transacted among the system 102 and the computing devices 110 and 111. Some platform data may be stored in the memory 106 or retrievable from the data store 108 and/or the computing devices 109, 110, 111, 112. For example, for each trip, the location of the origin and destination (e.g., transmitted by the computing device 111), the fee, and the time can be obtained by the system 102. The location may include GPS (Global Positioning System) coordinates.

[0040] In some embodiments, the system 102 and the one or more of the computing devices (e.g., the computing device 109) may be integrated in a single device or system.

Alternatively, the system 102 and the one or more computing devices may operate as separate devices. The data store(s) may be anywhere accessible to the system 102, for example, in the memory 106, in the computing device 109, in another device (e.g., network storage device)

coupled to the system 102, or another storage location (e.g., cloud-based storage system, network file system, etc.), etc. Although the system 102 and the computing device 109 are shown as single components in this figure, it is appreciated that the system 102 and the computing device 109 can be implemented as single devices or multiple devices coupled together. The system 102 may be implemented as a single system or multiple systems coupled to each other. In general, the system 102, the computing device 109, the data store 108, and the computing device 110, 111, and 112 may be able to communicate with one another through one or more wired or wireless networks (e.g., the Internet) through which data can be communicated. Various aspects of the system 100 are described below in reference to FIG. 2 to FIG. 6.

[0041] FIG. 2 illustrates a flowchart of an exemplary method 200 for information display, in accordance with various embodiments. The exemplary method 200 may be implemented by one or more components of the system 100 (e.g., the computing device 110). For example, a non-transitory computer-readable storage medium (e.g., the memory 116) may store instructions that, when executed by a processor (e.g., the processor 114), cause the computing device 110 (e.g., the processor 114) to perform the method 200. The operations of method 200 presented below are intended to be illustrative. Depending on the implementation, the exemplary method 200 may include additional, fewer, or alternative steps performed in various orders or in parallel.

[0042] In some embodiments, at step 210, it may be determined if status information of an interface of an application meets a cache condition. If it is determined that the status information of the interface of the application meets the cache condition, at step 211, information associated with the interface (e.g., currently displayed information, background process information associated with the currently displayed information) may be stored in a cache. The information associated with the interface may be cached entirely or partially. The information associated with the interface may comprise: at least a portion of currently displayed information on the interface and/or at least a portion of background information associated with the currently displayed information. For example, if a vehicle plate number is currently displayed on the interface while a vehicle registration number associated with the vehicle is not displayed, both the vehicle plate number and the vehicle registration number may be cached. The information associated with the interface may be stored as a page structure. For example, in an html format, the vehicle plate number may be associated with various other elements on the currently displayed page. The information associated with the interface may be stored as one or more components in the page structure. If it is determined that the status information of the interface of the application does not meet the cache condition, at step 212, the information associated with the interface may not be stored in the cache. In some embodiments, the cache condition may be that the status information meet a configuration (e.g., a currently displayed interface is a predetermined interface that triggers cache). For instance, an application may be configured such that certain interfaces it displays are tagged as important interfaces and are automatically saved to cache when displayed. Thus, when the application is in use, a mobile phone may constantly monitor the current interface of the application to determine if the current interface is one those tagged interfaces. If so, the mobile phone may store the current interface in a cache for a certain period of time. Further examples of the interfaces that meet the cache condition such as “order placed,” “boarding vehicle,” “in ride,” and “ride finished” interfaces are provided below with reference to FIGs. 3A to 3D. Such interfaces may be tagged in advance to trigger automatic caching when displayed.

[0043] At step 213, it is determined if a disruption to the application’s operation occurred. As described earlier, there may be various examples and reasons for the disruption. There are various ways to determine if the disruption occurred. For example, an event log on the computing device or the server may be searched to determine if there was a disruption to the normal operation of the application. If so, the computing device may determine that the

disruption occurred. Similarly, operating system log search or other methods may be used to determine if the disruption occurred.

[0044] In one example, the disruption comprises the application being killed. The application is restarted after being killed. At step 201, an application may be activated (e.g., started, restarted, auto-started, etc.). The operating system or another software of the computing device 110 may have activated the application. Alternatively, a user may have manually activated the application. Then, at step 202, it may be determined if the application was killed before the activation. The memory storage may be measured. If the available memory storage was below a threshold, it may be determined that the application was killed and thus the disruption occurred. Thus, in response to the activation of the application, determining if the cached application information of the interface exists in the cache comprises: in response to the activation of the application, determining if the application is killed; and in response to determining that the application is killed, determining if the cached application information of the interface exists in the cache.

[0045] In another example, the disruption includes a disruption to the application's connectivity with internet. Whether the disruption occurred can be determined based on the wireless connection condition. If the wireless connectivity or internet connectivity was below a threshold for more than a threshold time period, it may be determined that the disruption occurred. The connectivity may be measured based on various metrics, such as the rate of data packets transmitted and/or received.

[0046] At step 214, in response to no disruption to the application's operation, a current interface of the application may be displayed. For example, the application may have been exited or tabbed out normally. Thus, when the application is resumed, the last-displaying interface may be displayed.

[0047] At step 215, in response to the disruption to the application's operation, it may be determined if cached application information of the interface exists in the cache. The cache may include any type of temporary storage. In some embodiments, if the last displayed interface prior to the disruption corresponds to any of the tagged interfaces (matching the cache condition), the computing device may determine that the cached application information exists. Alternatively, the computing device may search its cache to determine if the cached application information corresponding to the last displayed interface prior to the disruption exists in the cache.

[0048] At step 216, in response to determining that the cached application information does not exist in the cache, a designated interface of the application may be loaded for display. In

some embodiments, the last displayed interface prior to the disruption is unimportant and not tagged (not matching the cache condition). In one example, the designated interface comprises a home page of the application. In another example, storing the information associated with the interface in the cache comprises storing the currently displayed information associated with a timestamp in the cache; and the designated interface comprises a cached interface with a latest timestamp.

[0049] At step 217, in response to determining that the cached application information exists in the cache, the cached application information may be loaded for display. In some embodiments, the last displayed interface prior to the disruption is important and tagged (matching the cache condition). Since loading data from cache is quick and efficient, the time for restoring the interface last displayed prior to the disruption can be significantly shortened. In some embodiments, loading the cached application information for display includes displaying the latest interface of the application, the latest interface includes the cached application information, and the cached application information is cached for a predetermined time period. For example, if the rides mostly complete in thirty minutes, the predetermined time period for cache may be set to one hour. Thus, with step 217, when the last page prior to the disruption needs to be shown, the cached application information can be loaded for display.

[0050] At step 218, the latest status of the interface (e.g., prior to the disruption or as updated after the disruption) may be requested for from a server (e.g., the system 102). At step 219, in response to obtaining the latest status from the server, the interface in the latest status may be restored for display. Thus, the interface displayed based on the cached application information may be fully restored to a normal interface of the application. For example, the command triggers in the interface can thus function formally. At step 220, in response to failing to obtain the current status from the server, a current interface of the application (e.g., an interface loaded from the cache) may be displayed, and an error notification may be triggered.

[0051] In some embodiments, after step 219, the method 200 may further include: obtaining a user input via a command trigger in the restored interface; and executing operations of the application based on the user input. The command trigger is described in further details below with reference to FIGs. 3A-3D. For example, if the command trigger is canceling an order, the executed operations may include canceling the order from the vehicle hailing platform.

[0052] FIGs. 3A-3D illustrate exemplary interfaces of an application for information display, in accordance with various embodiments. The illustrations shown in FIGs. 3A-3D and presented below are intended to be illustrative. As described below, the interfaces “order placed,” “boarding vehicle,” “in ride,” and “ride finished” may be examples of important
5 interfaces of the application that are configured to be cached when displayed. That is, such interfaces have met the cache condition. When such interfaces are displayed, the terminal device may automatically store the interface (e.g., application information in the displayed interface) in a cache or another temporary storage for a predetermined period of time (e.g., one hour). Such cache condition may be configured for the application. The interface that
10 meets the cache condition may be cached entirely or partially. For example, the specific parts of application information within the interface that need to be cached may be configured.

[0053] In some embodiments, the cache condition comprises the interface being an “order placed” interface, for which a ride hailing order has been placed through the application by a user, and the user has not started the ride. FIG. 3A illustrates such an exemplary interface
15 for information display, in accordance with various embodiments. As shown, a user has submitted an order on-line for ride hailing and may be waiting for the dispatched vehicle. The ordered ride is from an origin AAA to a destination BBB. A suggested route is highlighted. The dispatched vehicle/driver is 4 minutes, 1.1 km away from the pinned pick-up location. As shown, the cached application information may include: a map, order information, and/or
20 command triggers. The order information may include: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order (e.g., driver name), and/or vehicle information for the ride hailing order (e.g., a vehicle type, vehicle plate number). The command triggers may include: returning to a previous interface (e.g., a return button), initiating instant messenger communication with the
25 driver (e.g., a messaging button), initiating phone call with the driver (e.g., a call button), cancelling the ride hailing order (e.g., a cancel button), sharing the rider hailing order (e.g., a share button), and/or initiating communication with customer service (e.g., a help button). If the order status changes, cancelling the ride hailing order may trigger an error notification.

[0054] In some embodiments, the cache condition comprises the interface being a
30 “boarding vehicle” interface, for which a ride hailing order has been placed through the application by a user, and the user has not started the ride. For example, the user may be walking towards the vehicle or getting on the vehicle. FIG. 3B illustrates such an exemplary interface 320 for information display, in accordance with various embodiments. As shown,

the cached application information may include: a map, order information, and/or command triggers, which are similar to those shown in FIG. 3A.

[0055] In some embodiments, the cache condition comprises the interface being an “in ride” interface, for which a ride hailing order has been placed through the application by a user, and the user is riding in a vehicle that picked up the user and proceeds towards the destination. FIG. 3C illustrates such an exemplary interface 330 for information display, in accordance with various embodiments. As shown, the cached application information may include: a map, order information, and/or command triggers. The order information may include: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, and/or vehicle information for the ride hailing order. The command triggers may include: returning to a previous interface, initiating instant messenger communication with the driver, initiating phone call with the driver, showing a cost for the ride hailing order, sharing the rider hailing order, and/or initiating communication with customer service. If the ride is finished, the cost for the ride hailing order may be triggered to display details of the ride.

[0056] In some embodiments, the cache condition comprises the interface being a “ride finished” interface, for which a ride hailing order was placed through the application by a user, and the user has finished the ride. FIG. 3D illustrates such an exemplary interface 340 for information display, in accordance with various embodiments. As shown, the cached application information may include: a map and/or order information. The order information may include: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, vehicle information for the ride hailing order, and/or details of cost for the rider hailing order.

[0057] FIG. 4 illustrates a flowchart of an exemplary method 400 for information display, according to various embodiments of the present disclosure. The method 400 may be implemented in various environments including, for example, the system 100 of FIG. 1. The exemplary method 400 may be implemented by one or more components of the system 100 (e.g., the computing device 110). For example, a non-transitory computer-readable storage medium (e.g., the memory 116) may store instructions that, when executed by a processor (e.g., the processor 114), cause the computing device 110 (e.g., the processor 114) to perform the method 400. The operations of method 400 presented below are intended to be illustrative. Depending on the implementation, the exemplary method 400 may include additional, fewer, or alternative steps performed in various orders or in parallel.

[0058] Block 402 includes determining if status information of an interface of an application meets a cache condition. Block 404 includes, in response to determining that the status information meets the cache condition, storing information associated with the interface in a cache. Block 406 includes, in response to an activation of the application, determining if cached application information of the interface exists in the cache. Block 408 includes, in response to determining that the cached application information exists in the cache, loading the cached application information for display.

[0059] In some embodiments, the activation of the application comprises: a restart of the application.

[0060] In some embodiments, the application is restarted after being killed. In response to the activation of the application, determining if the cached application information of the interface exists in the cache comprises: in response to the activation of the application, determining if the application is killed; and in response to determining that the application is killed, determining if the cached application information of the interface exists in the cache.

[0061] In some embodiments, the method further comprises: requesting for a latest status of the interface from a server; in response to obtaining the latest status from the server, restoring the interface in the latest status for display; and in response to failing to obtain the current status from the server, displaying a current interface of the application and triggering error notification.

[0062] In some embodiments, before determining if the cached application information of the interface exists in the cache, the method further comprises: displaying a current interface of the application. For example, if there is no disruption to the application's operation (e.g., no reactivation), the application may continue displaying a current interface or otherwise performing normal function.

[0063] In some embodiments, the method further comprises: in response to determining that the cached application information of the interface does not exist, loading a designated interface of the application for display. In one example, the designated interface comprises a home page of the application. In another example, storing the information associated with the interface in the cache comprises storing the currently displayed information associated with a timestamp in the cache; and the designated interface comprises a cached interface with a latest timestamp.

[0064] In some embodiments, the information associated with the interface comprises: at least a portion of currently displayed information on the interface and/or at least a portion of

background information associated with the currently displayed information. The information associated with the interface is stored as a page structure.

[0065] The techniques described herein are implemented by one or more special-purpose computing devices. The special-purpose computing devices may be desktop computer systems, server computer systems, portable computer systems, handheld devices, networking devices or any other device or combination of devices that incorporate hard-wired and/or program logic to implement the techniques. Computing device(s) are generally controlled and coordinated by operating system software. Conventional operating systems control and schedule computer processes for execution, perform memory management, provide file system, networking, I/O services, and provide a user interface functionality, such as a graphical user interface (“GUI”), among other things.

[0066] FIG. 5 illustrates a block diagram of an exemplary computer system 500 for information display, in accordance with various embodiments. The system 500 may be an exemplary implementation of the computing device 110 of FIG. 1 or one or more similar devices. The method 400 may be implemented by the computer system 500. The computer system 500 may include one or more processors and one or more non-transitory computer-readable storage media (e.g., one or more memories) coupled to the one or more processors and configured with instructions executable by the one or more processors to cause the system or device (e.g., the processor) to perform the method 400. The computer system 500 may include various units/modules corresponding to the instructions (e.g., software instructions). In some embodiments, the computer system 500 may include a first determining module 502 configured to, determine if status information of an interface of an application meets a cache condition; a storing module 504 configured to, in response to determining that the status information meets the cache condition, store information associated with the interface in a cache; a second determining module 506 configured to, in response to an activation of the application, determine if cached application information of the interface exists in the cache; and a loading module 508 configured to, in response to determining that the cached application information exists in the cache, load the cached application information for display.

[0067] FIG. 6 is a block diagram that illustrates a computer system 600 upon which any of the embodiments described herein may be implemented. The system 600 may correspond to the computing device 110 or the system 500 described above. The computer system 600 includes a bus 602 or other communication mechanism for communicating information, one or more hardware processors 604 coupled with bus 602 for processing information. Hardware

processor(s) 604 may be, for example, one or more general purpose microprocessors. The processor(s) 604 may correspond to the processor 114 described above.

[0068] The computer system 600 also includes a main memory 606, such as a random access memory (RAM), cache and/or other dynamic storage devices, coupled to bus 602 for storing information and instructions to be executed by processor 604. Main memory 606 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 604. Such instructions, when stored in storage media accessible to processor 604, render computer system 600 into a special-purpose machine that is customized to perform the operations specified in the instructions.

The computer system 600 further includes a read only memory (ROM) 608 or other static storage device coupled to bus 602 for storing static information and instructions for processor 604. A storage device 610, such as a magnetic disk, optical disk, or USB thumb drive (Flash drive), etc., is provided and coupled to bus 602 for storing information and instructions. The main memory 606, the ROM 608, and/or the storage 610 may correspond to the memory 116 described above.

[0069] The computer system 600 may implement the techniques described herein using customized hard-wired logic, one or more ASICs or FPGAs, firmware and/or program logic which in combination with the computer system causes or programs computer system 600 to be a special-purpose machine. According to one embodiment, the techniques herein are performed by computer system 600 in response to processor(s) 604 executing one or more sequences of one or more instructions contained in main memory 606. Such instructions may be read into main memory 606 from another storage medium, such as storage device 610.

Execution of the sequences of instructions contained in main memory 606 causes processor(s) 604 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions.

[0070] The main memory 606, the ROM 608, and/or the storage 610 may include non-transitory storage media. The term “non-transitory media,” and similar terms, as used herein refers to a media that store data and/or instructions that cause a machine to operate in a specific fashion. The media excludes transitory signals. Such non-transitory media may include non-volatile media and/or volatile media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 610. Volatile media includes dynamic memory, such as main memory 606. Common forms of non-transitory media include, for example, a floppy disk, a flexible disk, hard disk, solid state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage medium, any

physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, NVRAM, any other memory chip or cartridge, and networked versions of the same.

[0071] The computer system 600 also includes a network interface 618 coupled to bus 602.

Network interface 618 provides a two-way data communication coupling to one or more

5 network links that are connected to one or more local networks. For example, network

interface 618 may be an integrated services digital network (ISDN) card, cable modem,

satellite modem, or a modem to provide a data communication connection to a corresponding

type of telephone line. As another example, network interface 618 may be a local area

network (LAN) card to provide a data communication connection to a compatible LAN (or

10 WAN component to communicate with a WAN). Wireless links may also be implemented.

In any such implementation, network interface 618 sends and receives electrical,

electromagnetic or optical signals that carry digital data streams representing various types of information.

[0072] The computer system 600 can send messages and receive data, including program

15 code, through the network(s), network link and network interface 618. In the Internet

example, a server might transmit a requested code for an application program through the

Internet, the ISP, the local network and the network interface 618.

[0073] The received code may be executed by processor 604 as it is received, and/or stored in storage device 610, or other non-volatile storage for later execution.

20 **[0074]** Each of the processes, methods, and algorithms described in the preceding sections

may be embodied in, and fully or partially automated by, code modules executed by one or

more computer systems or computer processors including computer hardware. The processes

and algorithms may be implemented partially or wholly in application-specific circuitry.

[0075] The various features and processes described above may be used independently of

25 one another, or may be combined in various ways. All possible combinations and sub-

combinations are intended to fall within the scope of this disclosure. In addition, certain

method or process blocks may be omitted in some implementations. The methods and

processes described herein are also not limited to any particular sequence, and the blocks or

states relating thereto can be performed in other sequences that are appropriate. For example,

30 described blocks or states may be performed in an order other than that specifically disclosed,

or multiple blocks or states may be combined in a single block or state. The exemplary blocks

or states may be performed in serial, in parallel, or in some other manner. Blocks or states

may be added to or removed from the disclosed exemplary embodiments. The exemplary

systems and components described herein may be configured differently than described. For

example, elements may be added to, removed from, or rearranged compared to the disclosed exemplary embodiments.

[0076] The various operations of exemplary methods described herein may be performed, at least partially, by an algorithm. The algorithm may be included in program codes or instructions stored in a memory (e.g., a non-transitory computer-readable storage medium described above). Such algorithm may include a machine learning algorithm. In some embodiments, a machine learning algorithm may not explicitly program computers to perform a function, but can learn from training data to make a predictions model that performs the function.

[0077] The various operations of exemplary methods described herein may be performed, at least partially, by one or more processors that are temporarily configured (e.g., by software) or permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors may constitute processor-implemented engines that operate to perform one or more operations or functions described herein.

[0078] Similarly, the methods described herein may be at least partially processor-implemented, with a particular processor or processors being an example of hardware. For example, at least some of the operations of a method may be performed by one or more processors or processor-implemented engines. Moreover, the one or more processors may also operate to support performance of the relevant operations in a “cloud computing” environment or as a “software as a service” (SaaS).

[0079] Any process descriptions, elements, or blocks in the flow diagrams described herein and/or depicted in the attached figures should be understood as potentially representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process. Alternate implementations are included within the scope of the embodiments described herein in which elements or functions may be deleted, executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those skilled in the art.

[0080] As used herein, the term “or” may be construed in either an inclusive or exclusive sense. Moreover, plural instances may be provided for resources, operations, or structures described herein as a single instance. Additionally, boundaries between various resources, operations, engines, and data stores are somewhat arbitrary, and particular operations are illustrated in a context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within a scope of various embodiments of the

present disclosure. In general, structures and functionality presented as separate resources in the exemplary configurations may be implemented as a combined structure or resource.

Similarly, structures and functionality presented as a single resource may be implemented as separate resources. These and other variations, modifications, additions, and improvements

5 fall within a scope of embodiments of the present disclosure as represented by the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

[0081] Although an overview of the subject matter has been described with reference to specific exemplary embodiments, various modifications and changes may be made to these
10 embodiments without departing from the broader scope of embodiments of the present disclosure. Such embodiments of the subject matter may be referred to herein, individually or collectively, by the term “invention” merely for convenience and without intending to voluntarily limit the scope of this application to any single disclosure or concept if more than one is, in fact, disclosed.

15 **[0082]** The embodiments illustrated herein are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed. Other embodiments may be used and derived therefrom, such that structural and logical substitutions and changes may be made without departing from the scope of this disclosure. The Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by
20 the appended claims, along with the full range of equivalents to which such claims are entitled.

CLAIMS:

1. A computer-implemented method for information display, comprising:
determining if status information of an interface of an application meets a cache
5 condition;
in response to determining that the status information meets the cache condition, storing
information associated with the interface in a cache;
in response to an activation of the application, determining if cached application
information of the interface exists in the cache; and
10 in response to determining that the cached application information exists in the cache,
loading the cached application information for display.

2. The method of claim 1, wherein the activation of the application comprises:
a restart of the application.

3. The method of claim 1, wherein, in response to the activation of the application,
determining if the cached application information of the interface exists in the cache
comprises:
in response to the activation of the application, determining if the application is killed;
20 and
in response to determining that the application is killed, determining if the cached
application information of the interface exists in the cache.

4. The method of claim 1, wherein:
25 the information associated with the interface comprises at least one of: at least a portion
of currently displayed information on the interface or at least a portion of background
information associated with the currently displayed information.

5. The method of claim 1, further comprising:
30 requesting for a latest status of the interface from a server;
in response to obtaining the latest status from the server, restoring the interface in the
latest status for display; and
in response to failing to obtain the current status from the server, displaying a current
interface of the application and triggering error notification.

6. The method of claim 1, before determining if the cached application information of the interface exists in the cache, further comprising:
displaying a current interface of the application.

5

7. The method of claim 1, further comprising:
in response to determining that the cached application information of the interface does not exist, loading a designated interface of the application for display.

10

8. The method of claim 7, wherein:
the designated interface comprises a home page of the application.

15

9. The method of claim 7, wherein:
storing the information associated with the interface in the cache comprises storing
currently displayed information associated with a timestamp in the cache; and
the designated interface comprises a cached interface with a latest timestamp.

20

10. The method of claim 1, wherein:
the information associated with the interface is stored as a page structure.

25

11. The method of claim 1, wherein:
the cache condition comprises the interface being an “order placed” or “boarding vehicle” interface, for both of which a ride hailing order has been placed through the application by a user, and the user has not started the ride;
the cached application information comprises at least one of: a map, order information, or command triggers;
the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, or vehicle information for the ride hailing order; and
the command triggers comprise at least one of: returning to a previous interface, initiating instant messenger communication with the driver, initiating phone call with the driver, cancelling the ride hailing order, sharing the rider hailing order, or initiating communication with customer service.

12. The method of claim 1, wherein:

the cache condition comprises the interface being an “in ride” interface, for which a ride hailing order has been placed through the application by a user, and the user is riding in a vehicle that picked up the user;

5 the cached application information comprises at least one of: a map, order information, or command triggers;

the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, or vehicle information for the ride hailing order; and

10 the command triggers comprise at least one of: returning to a previous interface, initiating instant messenger communication with the driver, initiating phone call with the driver, showing a cost for the ride hailing order, sharing the rider hailing order, or initiating communication with customer service.

13. The method of claim 1, wherein:

the cache condition comprises the interface being a “ride finished” interface, for which a ride hailing order was placed through the application by a user, and the user has finished the ride;

20 the cached application information comprises at least one of: a map or order information; and

the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, vehicle information for the ride hailing order, or details of cost for the rider hailing order.

25 14. A system for information display, comprising one or more processors and a non-transitory computer-readable storage medium coupled to the one or more processors, the storage medium storing instructions that, when executed by the one or more processors, cause the system to perform operations comprising:

30 determining if status information of an interface of an application meets a cache condition;

in response to determining that the status information meets the cache condition, storing information associated with the interface in a cache;

in response to an activation of the application, determining if cached application information of the interface exists in the cache; and

in response to determining that the cached application information exists in the cache, loading the cached application information for display.

5

15. The system of claim 14, wherein the activation of the application comprises:
a restart of the application.

16. The system of claim 14, wherein, in response to the activation of the application,
10 determining if the cached application information of the interface exists in the cache
comprises:

in response to the activation of the application, determining if the application is killed;
and

15 in response to determining that the application is killed, determining if the cached
application information of the interface exists in the cache.

17. The system of claim 14, wherein:

the information associated with the interface comprises at least one of: at least a portion
of currently displayed information on the interface or at least a portion of background
20 information associated with the currently displayed information.

18. The system of claim 14, wherein the operations further comprise:

requesting for a latest status of the interface from a server;

25 in response to obtaining the latest status from the server, restoring the interface in the
latest status for display; and

in response to failing to obtain the current status from the server, displaying a current
interface of the application and triggering error notification.

19. The system of claim 14, wherein, before determining if the cached application
30 information of the interface exists in the cache, the operations further comprise:

displaying a current interface of the application.

20. The system of claim 14, wherein the operations further comprise:

in response to determining that the cached application information of the interface does not exist, loading a designated interface of the application for display.

21. The system of claim 20, wherein:

5 the designated interface comprises a home page of the application.

22. The system of claim 20, wherein:

storing the information associated with the interface in the cache comprises storing currently displayed information associated with a timestamp in the cache; and

10 the designated interface comprises a cached interface with a latest timestamp.

23. The system of claim 14, wherein:

the information associated with the interface is stored as a page structure.

24. The system of claim 14, wherein:

15 the cache condition comprises the interface being an “order placed” or “boarding vehicle” interface, for both of which a ride hailing order has been placed through the application by a user, and the user has not started the ride;

the cached application information comprises at least one of: a map, order information,
20 or command triggers;

the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, or vehicle information for the ride hailing order; and

the command triggers comprise at least one of: returning to a previous interface,
25 initiating instant messenger communication with the driver, initiating phone call with the driver, cancelling the ride hailing order, sharing the rider hailing order, or initiating communication with customer service.

25. The system of claim 14, wherein:

30 the cache condition comprises the interface being an “in ride” interface, for which a ride hailing order has been placed through the application by a user, and the user is riding in a vehicle that picked up the user;

the cached application information comprises at least one of: a map, order information, or command triggers;

the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, or vehicle information for the ride hailing order; and

the command triggers comprise at least one of: returning to a previous interface,
5 initiating instant messenger communication with the driver, initiating phone call with the driver, showing a cost for the ride hailing order, sharing the rider hailing order, or initiating communication with customer service.

26. The system of claim 14, wherein:

10 the cache condition comprises the interface being a “ride finished” interface, for which a ride hailing order was placed through the application by a user, and the user has finished the ride;

the cached application information comprises at least one of: a map or order information; and

15 the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, vehicle information for the ride hailing order, or details of cost for the rider hailing order.

20 27. A non-transitory computer-readable storage medium coupled to one or more processors, the storage medium storing instructions that, when executed by the one or more processors, cause the one or more processors to perform operations comprising:

determining if status information of an interface of an application meets a cache condition;

25 in response to determining that the status information meets the cache condition, storing information associated with the interface in a cache;

in response to an activation of the application, determining if cached application information of the interface exists in the cache; and

30 in response to determining that the cached application information exists in the cache, loading the cached application information for display.

28. The storage medium of claim 27, wherein the activation of the application comprises: a restart of the application.

29. The storage medium of claim 27, wherein, in response to the activation of the application, determining if the cached application information of the interface exists in the cache comprises:

in response to the activation of the application, determining if the application is killed;

5 and

in response to determining that the application is killed, determining if the cached application information of the interface exists in the cache.

30. The storage medium of claim 27, wherein:

10

the information associated with the interface comprises at least one of: at least a portion of currently displayed information on the interface or at least a portion of background information associated with the currently displayed information.

31. The storage medium of claim 27, wherein the operations further comprise:

15

requesting for a latest status of the interface from a server;

in response to obtaining the latest status from the server, restoring the interface in the latest status for display; and

in response to failing to obtain the current status from the server, displaying a current interface of the application and triggering error notification.

20

32. The storage medium of claim 27, wherein, before determining if the cached application information of the interface exists in the cache, the operations further comprise: displaying a current interface of the application.

25

33. The storage medium of claim 27, wherein the operations further comprise:

in response to determining that the cached application information of the interface does not exist, loading a designated interface of the application for display.

34. The storage medium of claim 33, wherein:

30

the designated interface comprises a home page of the application.

35. The storage medium of claim 33, wherein:

storing the information associated with the interface in the cache comprises storing currently displayed information associated with a timestamp in the cache; and

the designated interface comprises a cached interface with a latest timestamp.

36. The storage medium of claim 27, wherein:

the information associated with the interface is stored as a page structure.

5

37. The storage medium of claim 27, wherein:

the cache condition comprises the interface being an “order placed” or “boarding vehicle” interface, for both of which a ride hailing order has been placed through the application by a user, and the user has not started the ride;

10 the cached application information comprises at least one of: a map, order information, or command triggers;

the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, or vehicle information for the ride hailing order; and

15 the command triggers comprise at least one of: returning to a previous interface, initiating instant messenger communication with the driver, initiating phone call with the driver, cancelling the ride hailing order, sharing the rider hailing order, or initiating communication with customer service.

20 38. The storage medium of claim 27, wherein:

the cache condition comprises the interface being an “in ride” interface, for which a ride hailing order has been placed through the application by a user, and the user is riding in a vehicle that picked up the user;

25 the cached application information comprises at least one of: a map, order information, or command triggers;

the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, or vehicle information for the ride hailing order; and

30 the command triggers comprise at least one of: returning to a previous interface, initiating instant messenger communication with the driver, initiating phone call with the driver, showing a cost for the ride hailing order, sharing the rider hailing order, or initiating communication with customer service.

39. The storage medium of claim 27, wherein:

the cache condition comprises the interface being a “ride finished” interface, for which a ride hailing order was placed through the application by a user, and the user has finished the ride;

the cached application information comprises at least one of: a map or order information;

5 and

the order information comprises at least one of: an origin of the ride hailing order on the map, a destination of the ride hailing order on the map, information of a driver for the ride hailing order, vehicle information for the ride hailing order, or details of cost for the rider hailing order.

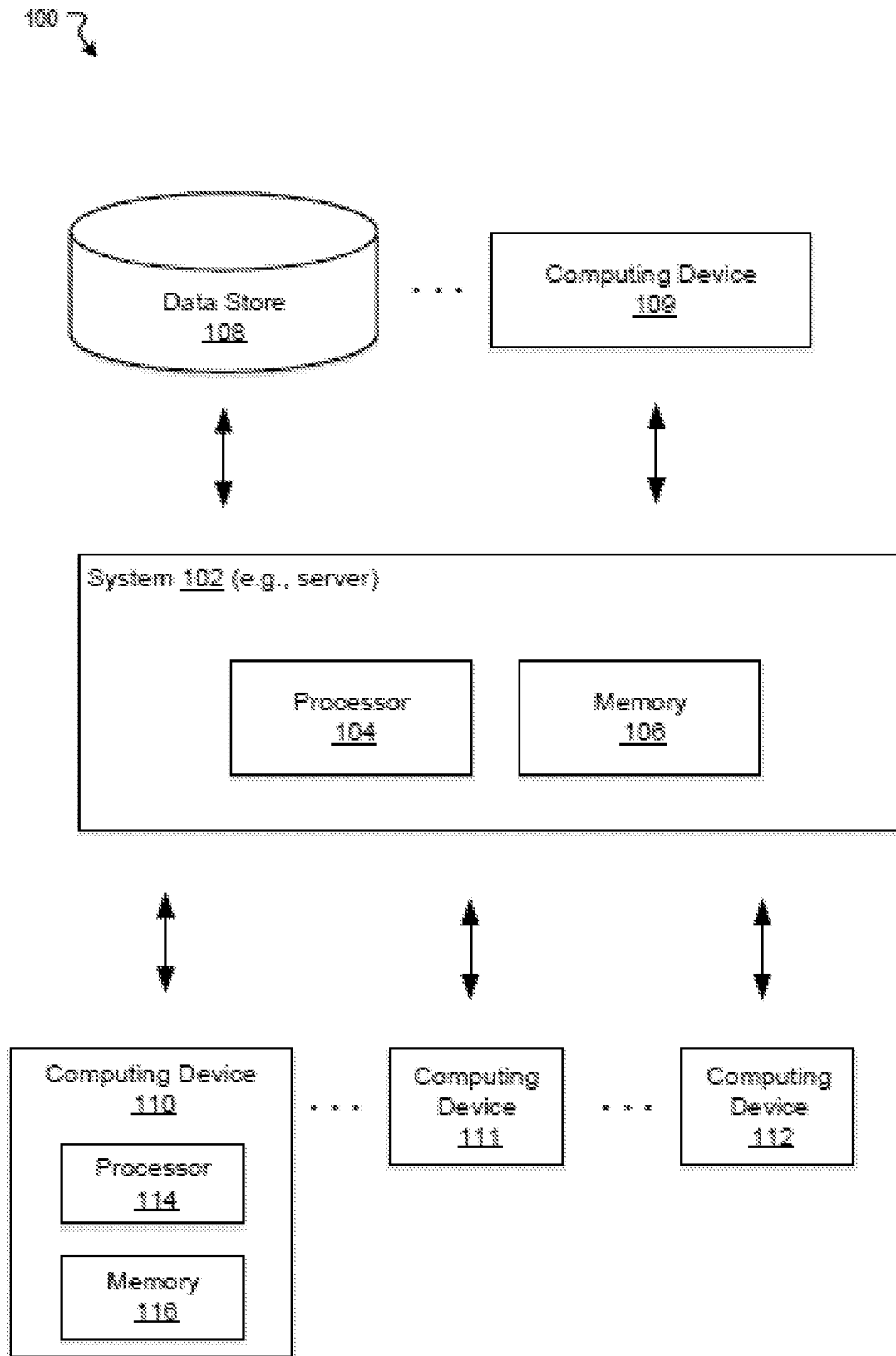


FIG. 1

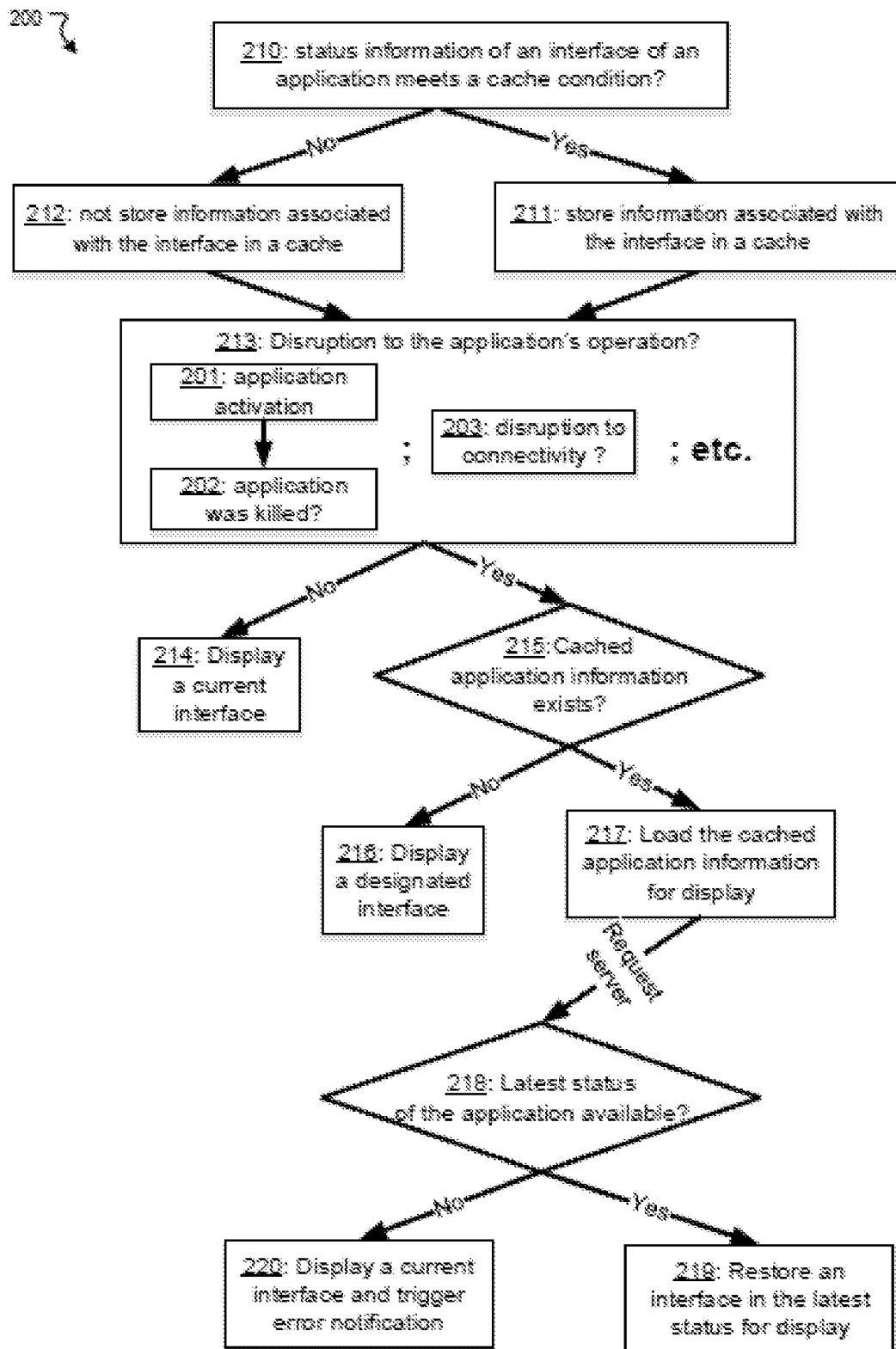


FIG. 2

310 ↗

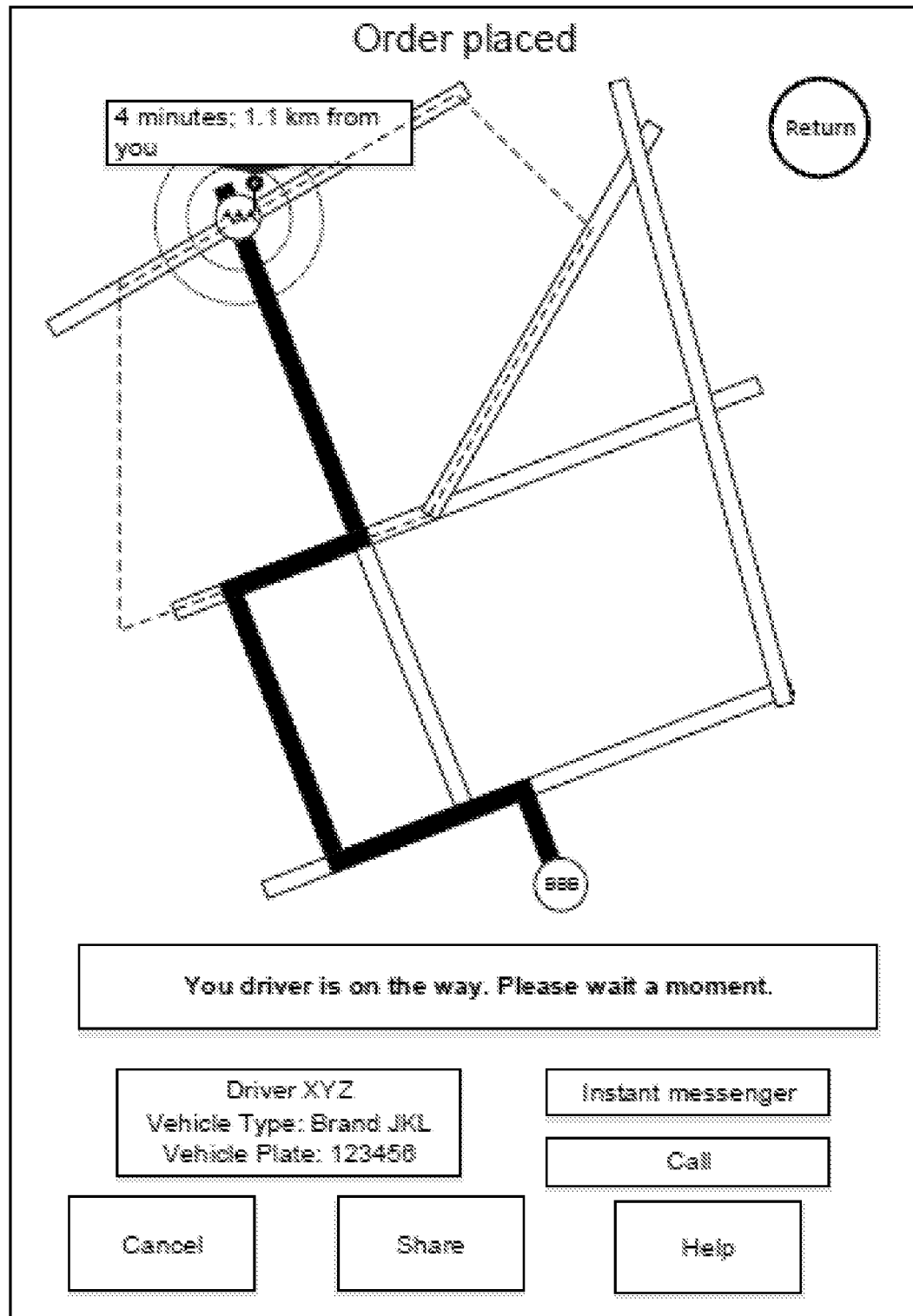


FIG. 3A

320

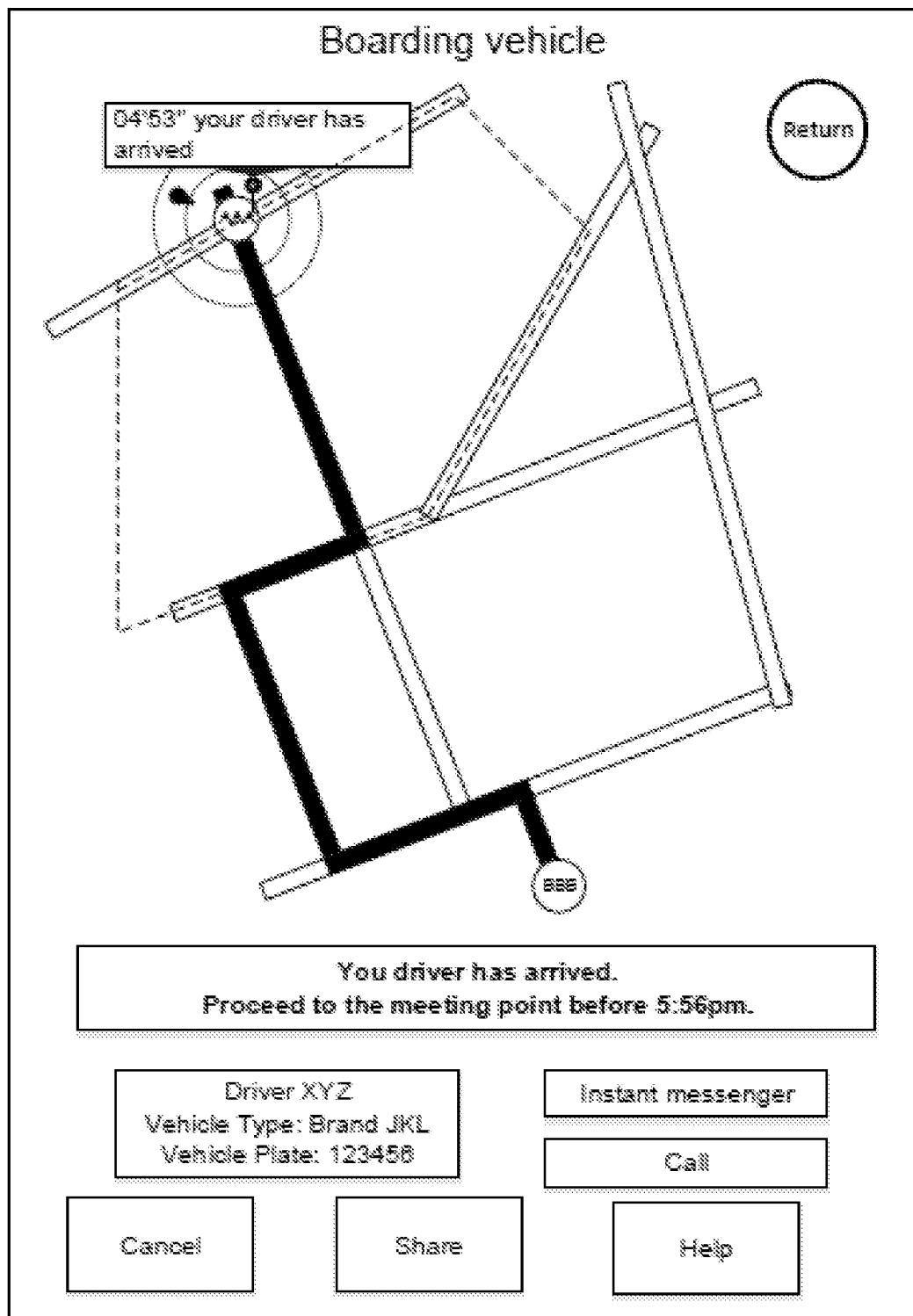


FIG. 3B

330 ↘

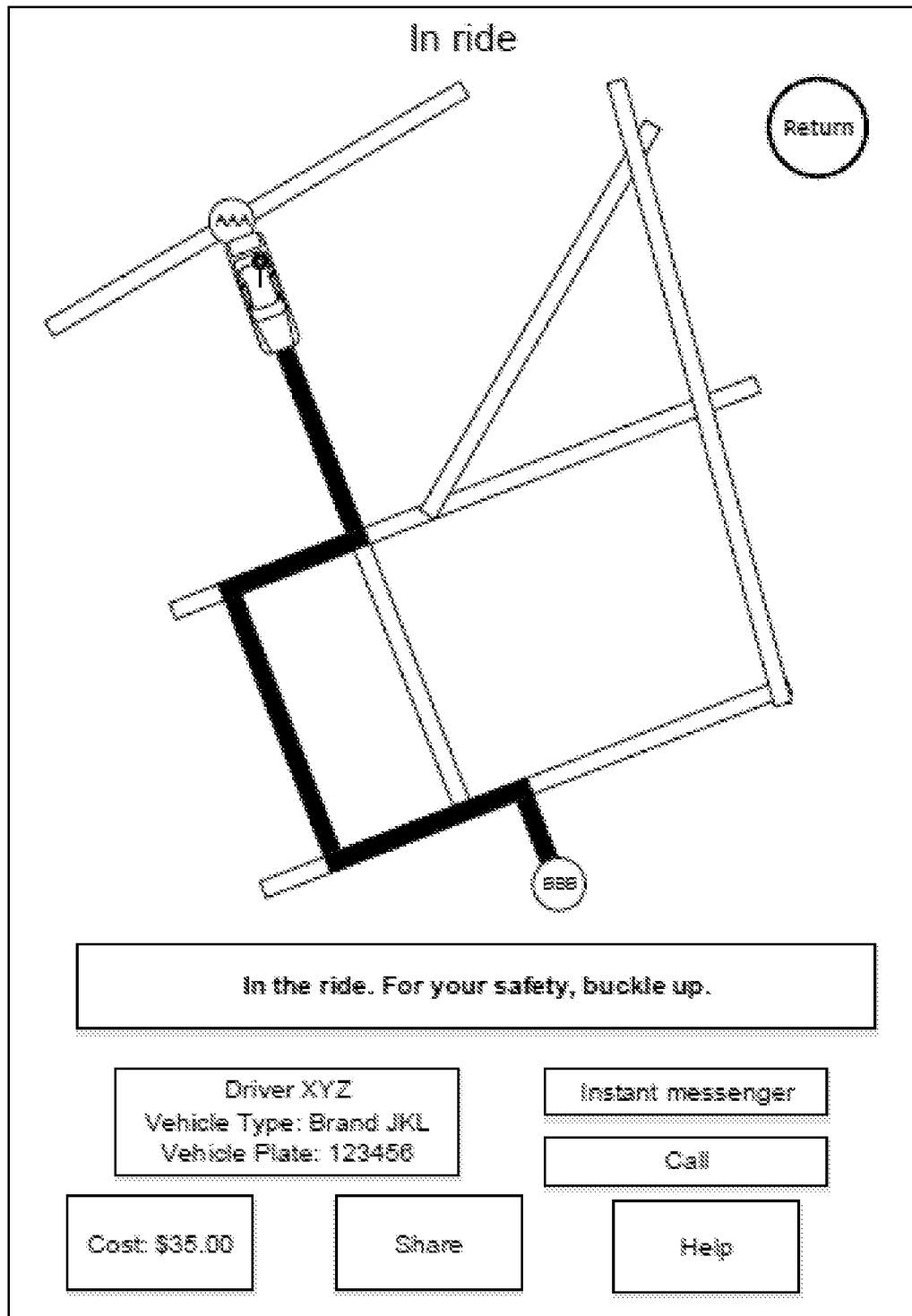


FIG. 3C

340 ↘

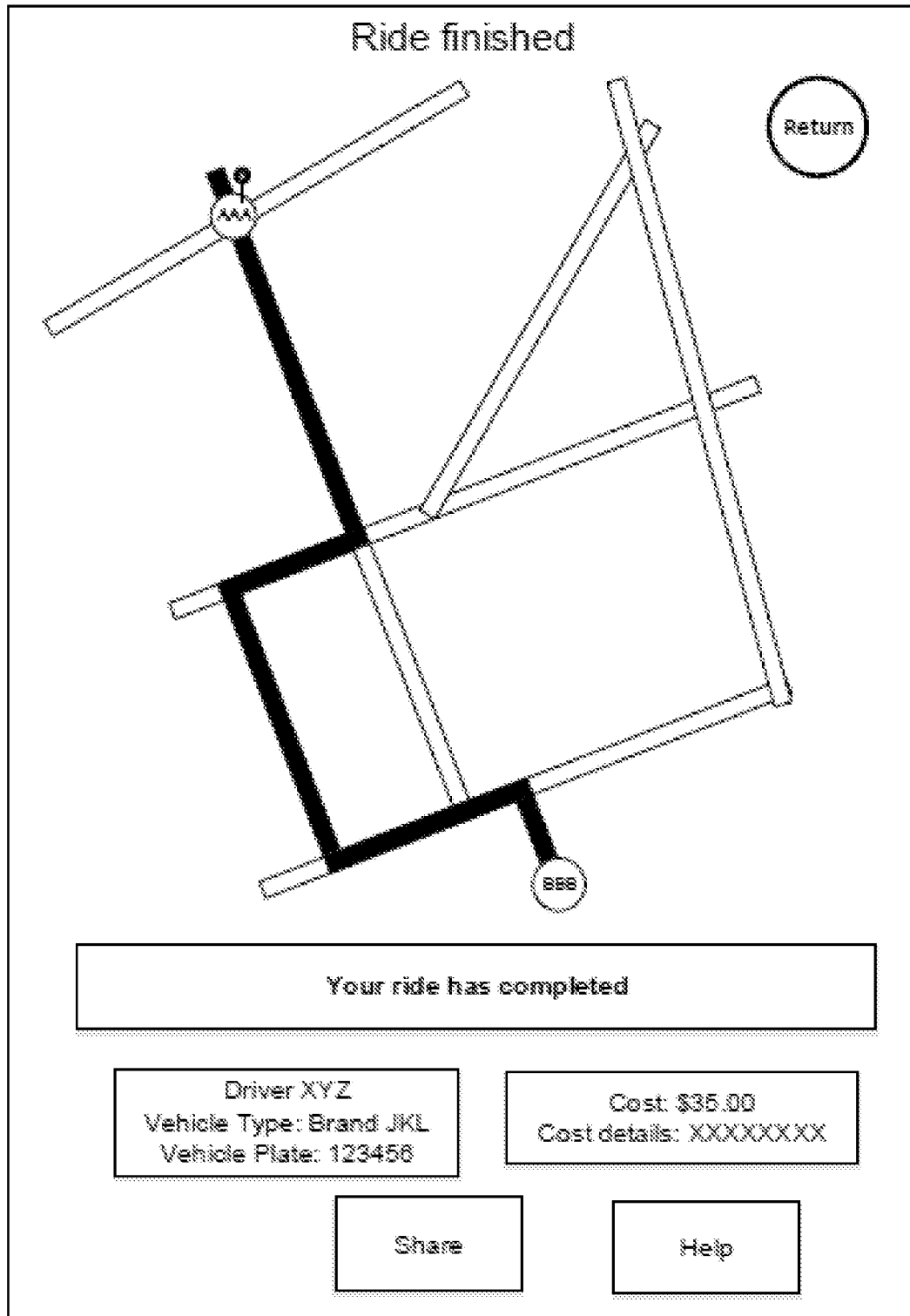


FIG. 3D

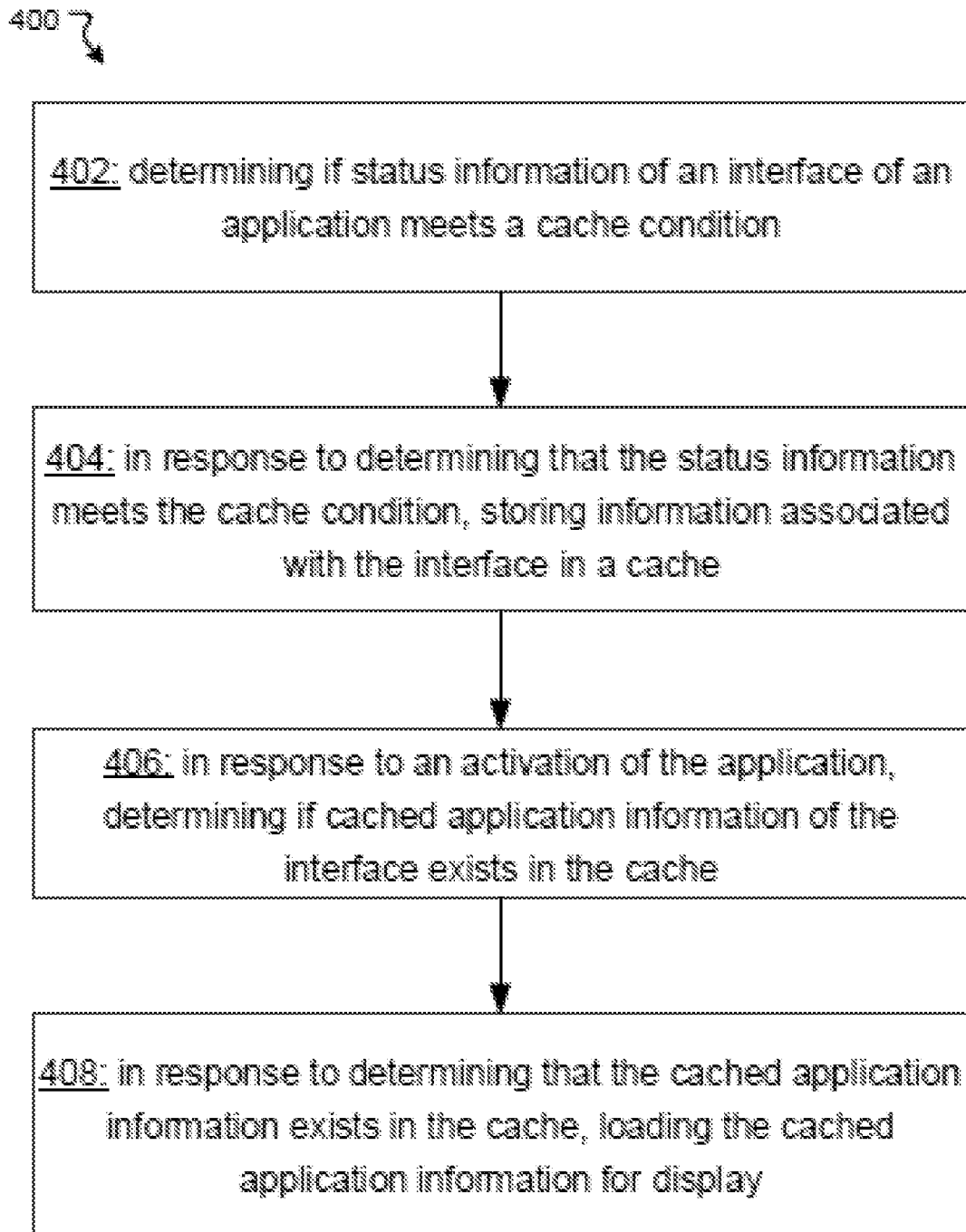


FIG. 4

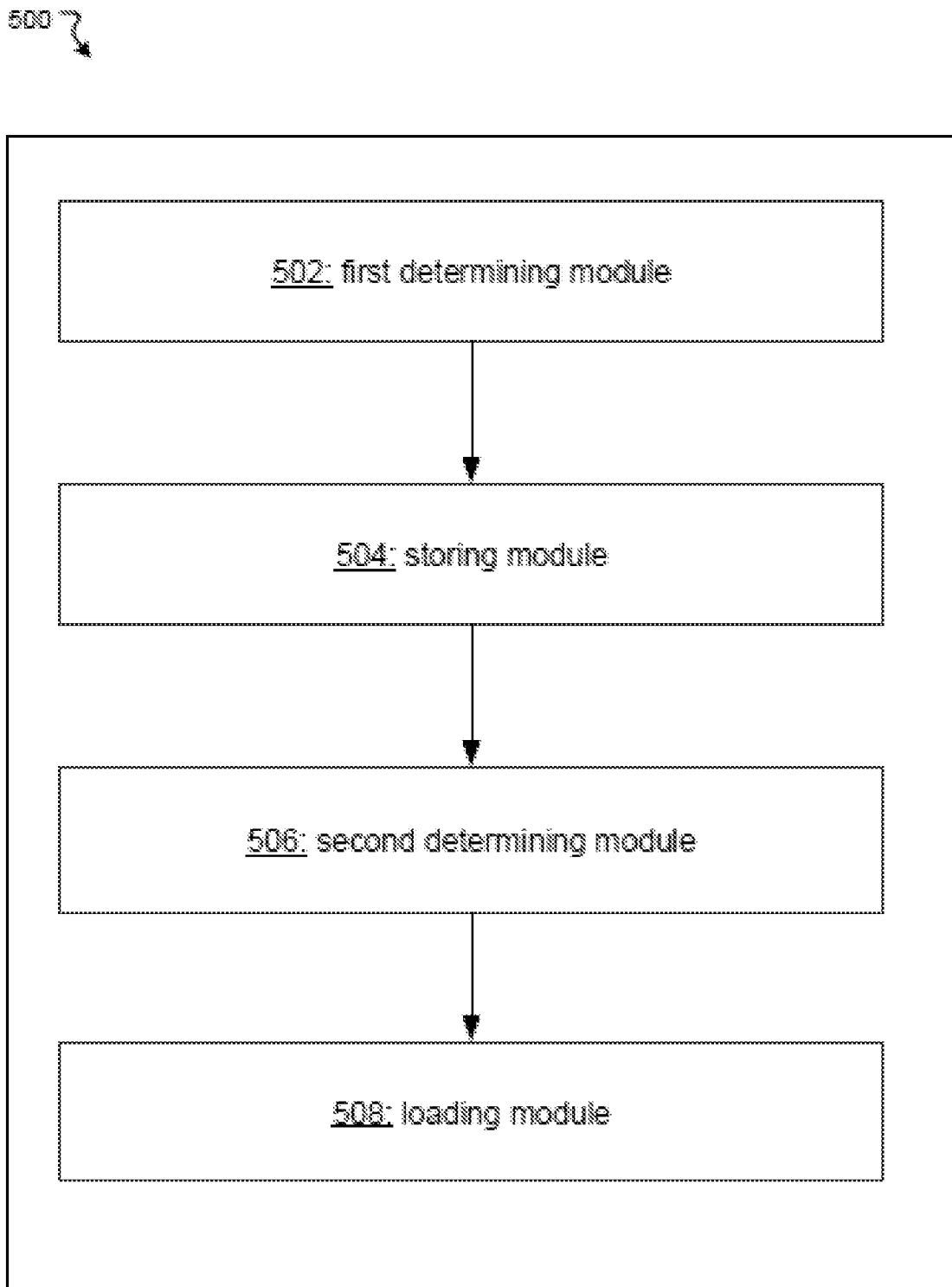


FIG. 5

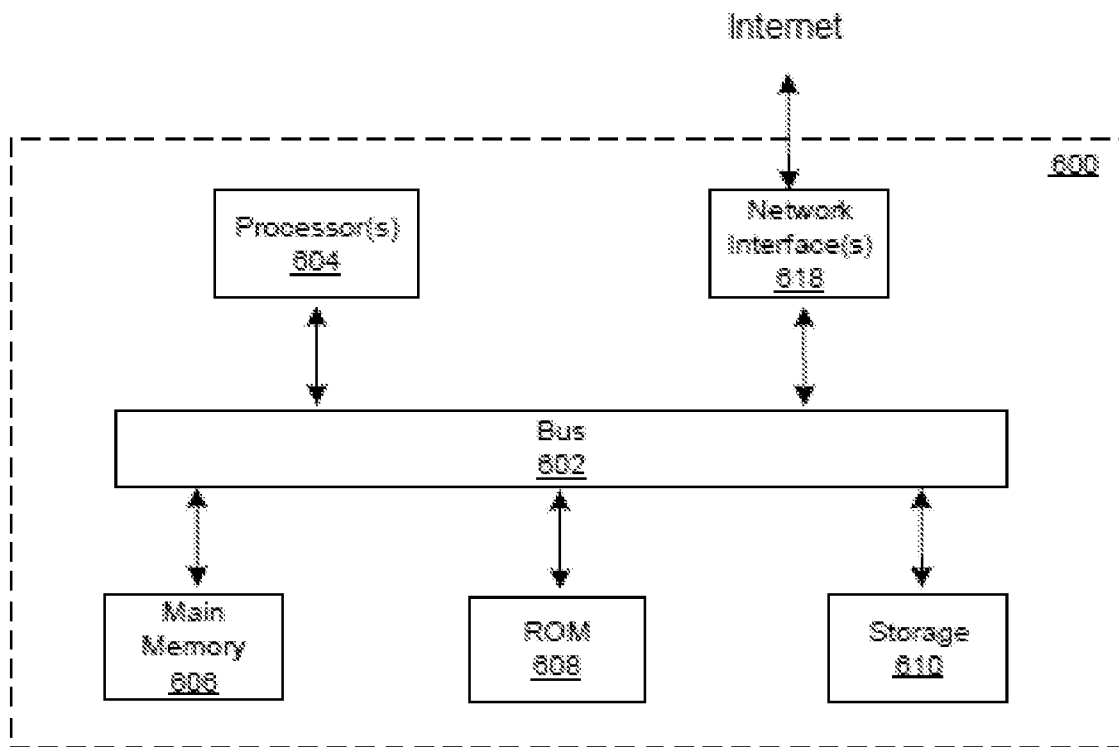


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/125485

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/451(2018.01)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)

CNPAT, WPI, EPODOC, CNKI, IEEE, GOOGLE: interface, GUI, application, APP, cache, buffer, restore, recover, restart, activation, kill

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 105045728 A (INSPUR GENERAL SOFTWARE CO., LTD.) 11 November 2015 (2015-11-11) description, paragraphs [0002]-[0009]	1-39
A	CN 102131152 A (ZTE CORPORATION) 20 July 2011 (2011-07-20) the whole document	1-39
A	CN 104618471 A (TIANJIN UNIVERSITY) 13 May 2015 (2015-05-13) the whole document	1-39
A	JP 2015190785 A (CANVAS MAPPLE CO., LTD.) 02 November 2015 (2015-11-02) the whole document	1-39

☐ 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

27 August 2019

Date of mailing of the international search report

26 September 2019

Name and mailing address of the ISA/CN

National Intellectual Property Administration, PRC
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing
100088
China

Authorized officer

SUN,Taomin

Facsimile No. (86-10)62019451

Telephone No. 86-(10)-53961420

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2018/125485

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN	105045728	A	11 November 2015	None	
CN	102131152	A	20 July 2011	None	
CN	104618471	A	13 May 2015	None	
JP	2015190785	A	02 November 2015	None	