May 2022

OPERATING SYSTEM-LEVEL SURVEY PLATFORM

Krishna Shukla

OPERATING SYSTEM-LEVEL SURVEY PLATFORM

ABSTRACT

An operating system (OS) of a computing device (e.g., a smartphone, mobile phone, a tablet computer, a laptop computer, a wearable device, etc.) may provide a survey platform that applications installed at the computing device can use to generate customizable, general-purpose surveys. For instance, the OS may allow the applications to perform a library call or a remote procedure call (RPC) to generate a customizable survey graphical user interface (GUI), which may be overlay-based, notification-based, etc. In some examples, the OS may control (e.g., restrict, limit, regulate, etc.) usage of the survey platform by the applications based on OS-wide permissions and OS-controlled app-specific permissions. In this way, a user of the computing device may be able to configure various survey parameters, such as the number of surveys any application can send in a given time period (e.g., a day, a week, a month, a year, etc.). To incentivize completion of the surveys, the survey platform may invoke an open-source application programming interface (API) that rewards users based on application configurations. For example, the API may reward a user with an application store credit, a free subscription to a digital product, etc., for completing the survey.

DESCRIPTION

FIG. 1 below shows an example computing device 100 that provides operating system-level survey capabilities in accordance with the techniques of this disclosure. Examples of computing device 100 include a mobile phone, a tablet computer, a laptop computer, a desktop computer, a server, a mainframe, a set-top box, a television, a wearable device (e.g., a computerized watch, computerized eyewear, computerized gloves, etc.), a home automation

device or system (e.g., an intelligent thermostat or home assistant device), a gaming system, a media player, an e-book reader, a mobile television platform, an automobile navigation or infotainment system, or any other type of mobile, non-mobile, wearable, and non-wearable computing device. As shown in FIG. 1, computing device 100 includes a presence-sensitive display 102 ("display 102"), one or more processors 104, one or more communication components 106, and one or more storage devices 108. As further shown in FIG. 1, storage devices 108 includes an operating system 110 ("OS 110") and one or more applications 112.

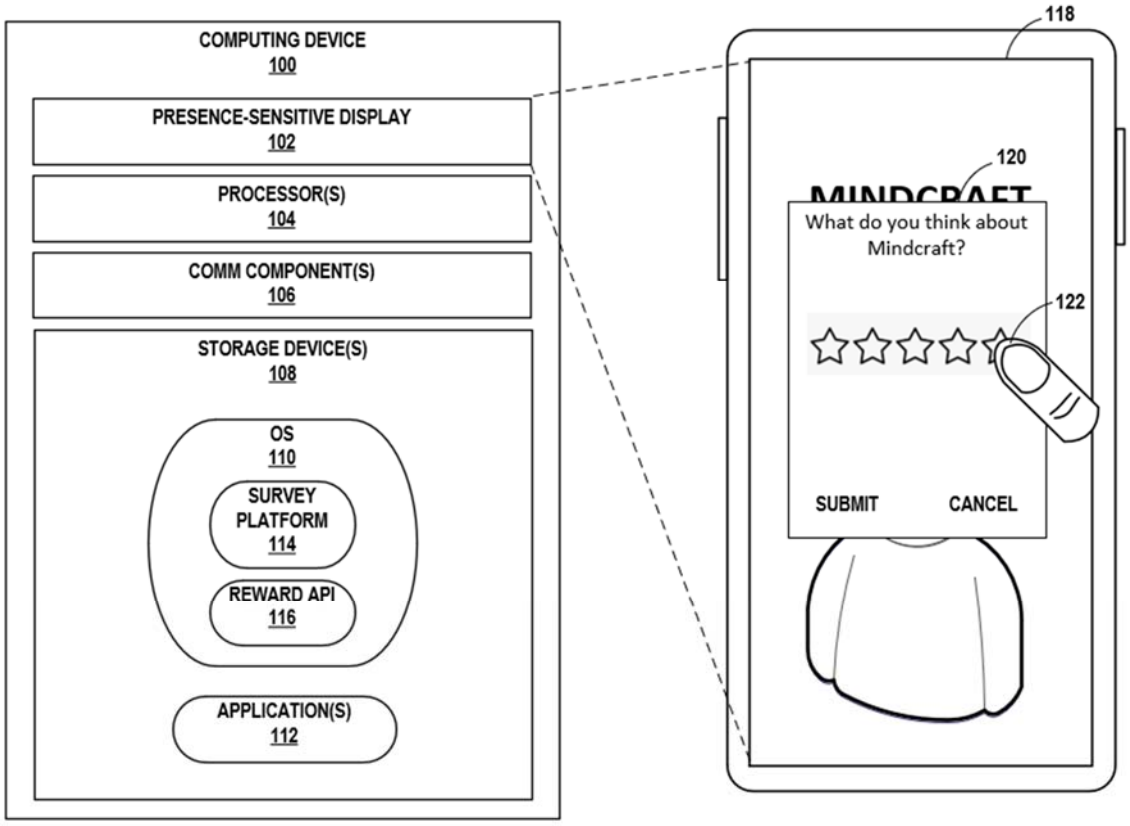

FIG. 1

Display 102 may function as an input device for computing device 100 using a touchscreen, pressure sensitive screen, an acoustic pulse recognition touchscreen or another

presence-sensitive screen technology. Display 102 may function as an output device for computing device 100 using any one or more of display technologies (e.g., liquid crystal display, dot matrix display, light emitting diode (LED) display, organic light emitting diode (OLED) display, electronic ink display, or similar display technology.

Processors 104 may implement functionality and/or execute instructions associated with computing device 100. Examples of processors 104 include one or more of an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), an application processor, a display controller, an auxiliary processor, a central processing unit (CPU), a graphics processing unit (GPU), one or more sensor hubs, and any other hardware configure to function as a processor, a processing unit, or a processing device.

Computing device 100 may include COMM components 106. COMM components 106 may receive and transmit various types of information over a network, such as a cellular radio, a third-generation (3G) radio, a fourth-generation (4G) radio, a fifth-generation (5G) radio, a Bluetooth® radio (or any other personal area network (PAN) radio), a near-field communication (NFC) radio, a WiFi® radio (or any other wireless local area network (WLAN) radio), and/or the like.

Storage devices 108 may include one or more computer-readable storage media. For example, storage devices 108 may be configured for long-term, as well as short-term storage of information, such as instructions, data, or other information used by computing device 100. In some examples, storage devices 108 may include non-volatile storage elements. Examples of such non-volatile storage elements include magnetic hard disks, optical discs, solid state discs, and/or the like. In other examples, in place of, or in addition to the non-volatile storage elements, storage devices 108 may include one or more so-called “temporary” memory devices, meaning

that a primary purpose of these devices may not be long-term data storage. For example, the devices may comprise volatile memory devices, meaning that the devices may not maintain stored contents when the devices are not receiving power. Examples of volatile memory devices include random-access memories (RAM), dynamic random-access memories (DRAM), static random-access memories (SRAM), etc.

Applications 112 may include messaging applications, social media applications, calendar applications, gaming applications, digital assistant applications, mapping applications, word processing applications, keyboard applications, application extensions, or any other type of applications configured to execute at computing device 100. Applications 112 may be cloud-based applications that are accessible from computing device 100 but execute at a remote server in a cloud network environment. Applications may generate GUIs, such as an application GUI 118.

To obtain feedback from the user of applications 112, developers of applications 112 may design applications 112 to send surveys querying about various aspects of applications 112, such as an overall rating (e.g., 1 to 5 stars), ease of use, consistency, etc. Because these surveys are created by different developers, the surveys may not only differ in substance, but also in quality. This lack of uniformity among surveys may frustrate the user and result in inaccurate responses to the surveys, which may make analysis of survey responses difficult. In addition, the user may be unable to globally control (e.g., at an OS-level) the ability of applications 112 to send the surveys in the first place, which may irritate the user if the user feels the user is being spammed with surveys.

In accordance with techniques of this disclosure, OS 110 of computing device 100 may provide survey platform 114 that applications 112 can use to generate customizable, general-

purpose surveys. In other words, applications 112 may interface with survey platform 114 to generate a survey GUI 120 with which a user can interact to complete a survey. In some examples, OS 110 may control (e.g., restrict, limit, regulate, etc.) usage of survey platform 114 by applications 112 based on OS-wide permissions and OS-controlled app-specific permissions. In this way, a user of the computing device may be able to configure various survey parameters, such as the number of surveys applications 112 can send in a given time period (e.g., a day, a week, a month, year, etc.).

As noted above, applications 112 may interface with survey platform 114 to generate survey GUI 120. In some examples, survey platform 114 may perform a library call (e.g., a request to use a specific function that is defined in a programming library) to generate survey GUI 120. For instance, responsive to one of applications 112 interfacing with survey platform 114 to generate survey GUI 120, survey platform 114 may request to use a survey generation function defined in a programming library provided by OS 110. OS 110 may grant the request from survey platform 114 to use the survey generation function, resulting in display 102 displaying survey GUI 120.

In other examples, survey platform 114 may use a remote procedure call (RPC) to generate survey GUI 120. As used here, RPC refers to a technique based on extending the conventional local procedure calling such that the called procedure need not exist in the same address space as the calling procedure. For example, a remote server may send a RPC to survey platform 114. If correctly authenticated, the RPC from the server may specify a survey to show. Survey platform 114 may then generate a survey per the RPC, resulting in display 102 displaying survey GUI 120.

The form (and optionally the content) of survey GUI 120 may change based on whether the application sending the survey is in the foreground (e.g., display 102 is displaying application GUI 118) or background (e.g., the application sending the survey is running, but display 102 is not displaying application GUI 118) of computing device 100. For example, when the application sending the survey is in the foreground of computing device 100, survey GUI 120 may be an overlay-based GUI (e.g., survey GUI 120 is displayed in front of application GUI 118). When the application sending the survey is in the background computing device 100, survey GUI 120 may be a notification-based GUI (e.g., survey GUI 120 is displayed as a notification). The notification-based version of survey GUI 120 may include fewer features than the overlay-based version of survey GUI 120. In some examples, a user may select the notification-based version of survey GUI 120 to cause computing device 100 to display the overlay-based version of survey GUI 120.

Surveys generated by survey platform 114 may be substantially similar (by virtue of being byproducts of the same survey generation function) in terms of form, content, quality, features, etc. Nonetheless, developers of applications 112 may customize various aspects of the surveys. For instance, a developer may program an application to provide survey platform 114 with parameters relating to colors, fonts, questions, when the survey is to appear (e.g., when a user opens an application), etc., which survey platform 114 may include in the library call or RPC. Survey platform 114 may include other parameters in the library call or RPC, such as the display resolution of computing device 100, whether the computing device 100 is set to “dark” or “light” mode, etc.

Survey platform 114 may enable customization of the questions of the survey. For example, survey platform 114 may, based on input from a developer, generate a rating question

(e.g., “How would you rate your experience with this application out of 5 stars?”), a multiple-choice question, an open response question, etc. In some examples, a developer may use survey platform 114 to make a question of a survey appear based on a custom condition (e.g., a specific answer to a previous question). For example, if, in response to a rating question, a user rates the application negatively (e.g., 3 stars or fewer), survey platform 114 may update survey GUI 120 to include a follow-up question such as “What could be improved from the multiple choices below?”, “Please expand what you would like to see in the open response below,” etc.

Based on these parameters, the survey generation function may generate customized surveys for applications 112, providing developers with the flexibility necessary to design unique products while maintaining an overarching theme for the surveys. For example, the survey generation function may select the dimensions of an answer field included in survey GUI 120 based on the screen resolution parameter. In another example, the survey generation function may select the size of the font of questions included in survey GUI 120 based on the font parameter. In yet another example, the survey generation function may select the color palette of survey GUI 120 based on whether the computing device 100 is set to “dark” or “light” mode.

In some examples, survey platform 114 of OS 110 may control (e.g., restrict, limit, regulate, etc.) usage of survey platform 114 by applications 112 based on OS-wide permissions and OS-controlled app-specific permissions. A user of computing device 100 may set the OS-wide permissions and OS-controlled app-specific permissions (e.g., by providing user input granting, denying, or otherwise selecting OS-wide permissions and OS-controlled app-specific permissions). For example, a user of computing device 100 may provide user input limiting the frequency that a particular application can interface with survey platform 114 to generate a survey. In another example, the user of computing device 100 may provide user input limiting

the frequency that all applications can interface with survey platform 114 to generate a survey. In yet another example, the user of computing device 100 may provide user input completely preventing some of applications 112 from interfacing with survey platform 114. Thus, by enabling the user to select OS-wide permissions and OS-controlled app-specific permissions, the user of computing device 100 can control, at an OS-level, the ability of applications 112 to send surveys, reducing the likelihood that the user will feel like the user is being spammed with unwanted surveys.

In any case, a user may complete the surveys generated by survey platform 114, and applications 112 may process (e.g., store, send, etc.) the responses to the surveys. For instance, the user may provide a user input 122 (e.g., via display 102) to select answers to questions included in survey GUI 120. Responsive to the user completing the survey, the application associated with the survey may send the survey response to the developer of the application for analysis. In examples where the survey is a review, the application may send the survey to the application store so that the survey response may be aggregated with other reviews by users of the application.

As shown in FIG. 1, OS 110 includes a reward application programming interface 116 ("reward API 116"). In general, an API is a set of definitions and protocols for building and integrating application software. Thus, reward API 116 may represent a set of definitions and protocols that rewards a user of computing device 100 for completing surveys sent by applications 112 via survey platform 114. Reward API 116 may provide rewards to users who complete the surveys based on application configurations. For example, responsive to completion of a survey sent by applications 112 via survey platform 114 and based on the application configurations, reward API 116 may reward (e.g., administer, provide, give, etc.) a

user with an application store credit, a free subscription to a digital product, etc., for completing the survey. Reward API 116 may be open source such that developers may freely inspect, review, and use reward API 116.

The techniques of this disclosure include one or more advantages. For example, by interfacing with survey platform 114 to call the survey generation function defined in the programming library, applications 112 may gain the survey generation functionality without implementing that behavior themselves, increasing software development efficiency and facilitating consistent user experiences with the surveys. In addition, the techniques enable the user to select OS-wide permissions and OS-controlled app-specific permissions, reducing the likelihood that the user will feel like the user is being spammed with unwanted surveys. Moreover, to incentivize completion of the surveys, developers of applications 112 may use reward API 116 to reward users who complete the survey with application store credits, free subscriptions to a digital product, etc., based on application configurations.

It is noted that the techniques of this disclosure may be combined with any other suitable techniques or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Publication No. US20180285102A1. In another example, the techniques of this disclosure may be combined with the techniques in U.S. Patent Application No. 2013/0111323A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application No. 2011/0231226A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in Dallas Thomas, “Earn Free Google Play Credits on Android by Filling Out Surveys,” Android Gadget Hacks, May 14, 2016. In yet another example, the

techniques of this disclosure may be combined with the techniques described in U.S. Patent Application No. 2021/0241296A1.