EXHIBIT F

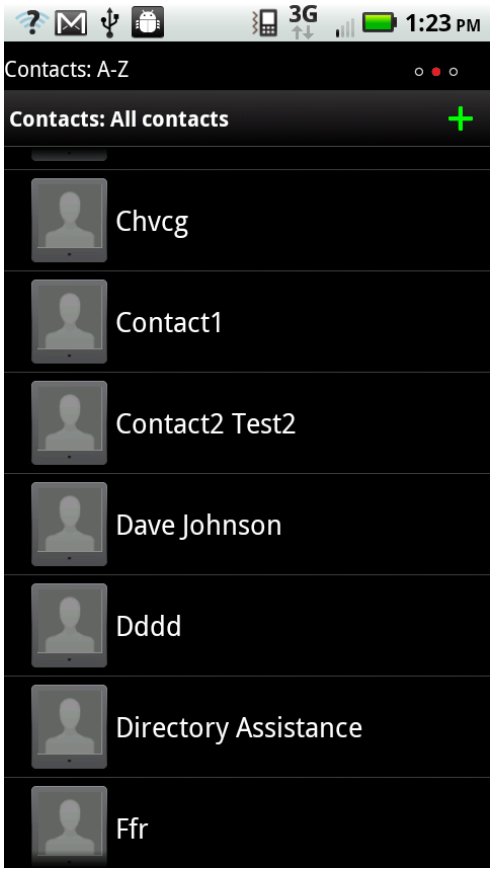
‘536 Patent Infringement Contentions

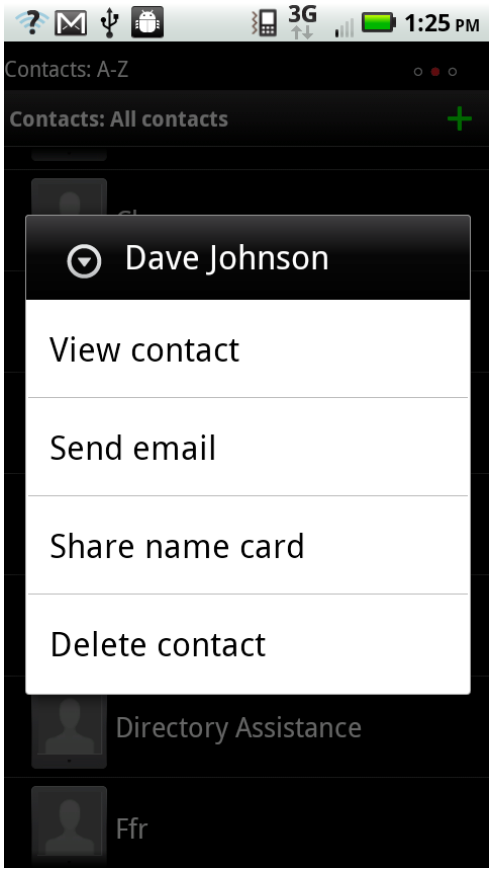
Motorola’s infringing products (“Accused Devices”) include mobile devices, such as smartphones, associated software, and components thereof. The Accused Devices include Motorola’s Android based phones which include, but are not limited to, the Motorola Droid X, Droid 2, Droid 2 Global, Cliq 2, Defy, Bravo, Droid Pro, Droid 2 R2-D2, Droid X 2, Charm, Droid, Flipside, Flipout, Atrix, Droid Bionic, Xoom, Devour A555, Backflip, Cliq/Dext, Cliq XT/Quench, Citrus, Spice, i1 and other Motorola Android based phones incorporating hardware and/or software that is substantially similar. The figures and illustrations in the infringement chart below display exemplary devices.

U.S. Patent No. 6,791,536 (‘536 Patent)	**Accused Devices**
14. A computer configured to simulate at least one gesture of a pointing device having a primary switch and a secondary switch responsive to stylus input, the computer comprising:	**Each Accused Device is a computer configured to simulate at least one gesture of a pointing device having a primary switch and a secondary switch responsive to stylus input**. Each Accused Device is a handheld computer. (*See* http://www.motorola.com/Consumers/US-EN/Consumer-Product-and-Services/Mobile-Phones/ci.Motorola-DROID-2-US-EN.alt) (describing Droid 2’s processor and other specifications.) Each Accused Device includes the Android operating system platform (“Android”). (*See id*.) By way of example, Figure 14-1 shows the Motorola device branded as Droid 2 (hereinafter, “Droid 2”).

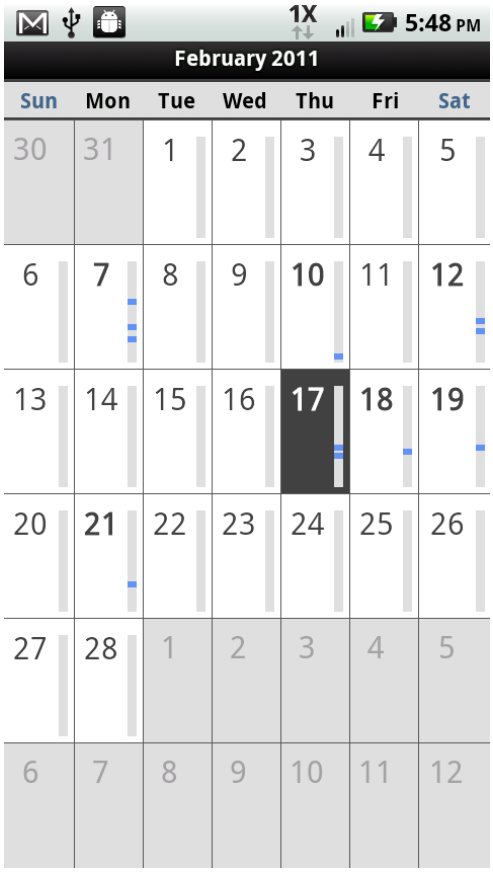
U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 14-1. Each Accused Device is configured to simulate at least one gesture of a pointing device having a primary switch and a secondary switch. Each Accused Device is configured to simulate a mouse. Such a mouse may have a "left" mouse button that serves as a primary switch and a "right" mouse button that serves as a secondary switch.
a touch-sensitive display surface;	**Each Accused Device has a touch-sensitive display surface**. Each Accused Device includes a touch-sensitive display. (*See id.*) (describing the touch screen interface of the Droid 2)
and a processor coupled to the touch-sensitive display surface and configured to detect whether the stylus is held against the touch-sensitive display surface for at least a threshold amount of time,	**Each Accused Device includes a processor coupled to the touch-sensitive display. This processor is configured to detect whether the stylus is held against the touch-sensitive display surface for at least a threshold amount of time**. Each Accused Device includes a processor. (*See id.*) (describing Motorola's Droid 2's computer processor and other specifications.) The processor is configured to detect whether a user presses his or her finger against the touch screen display for at least a threshold amount of time. The Android operating system on Each Accused Device provides a *GestureDetector* class for identifying "gestures." (http://developer.android.com/intl/zh-TW/reference/android/view/GestureDetector.html.) *GestureDetector* analyzes user input events and determines which gestures the user has performed. In one exemplary usage, *GestureDetector* detects "tap" and "long press" gestures. For a

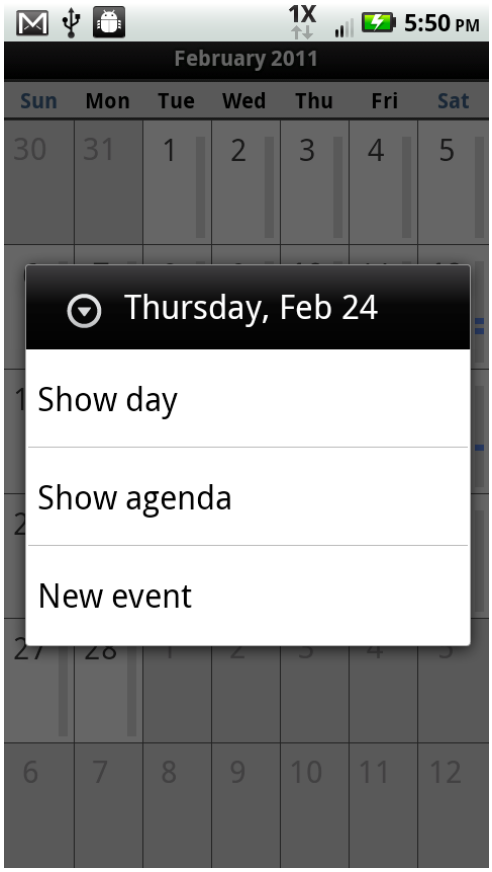
U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	"tap" gesture, the user's finger is held against the touch screen for less than a threshold amount of time and for a "long press" gesture, the user's finger is held against the touch screen for at least a threshold amount of time. The threshold is identified as a constant in the *GestureDetector* class. *(See* android/frameworks/base/core/java/android/view/GestureDetector.java)
and in response to the stylus being held against the touch-sensitive display surface for at least the threshold amount of time generating at least one event representing an activation of the secondary switch of the pointing device, and	Each Accused Device, in response to the stylus being held against the touch-sensitive display surface for at least the threshold amount of time, generates at least one event representing an activation of the secondary switch of the pointing device. As discussed above, *GestureDetector* determines when a user has performed a "long press" gesture. The nested class *GestureDetector.OnGestureListener* then generates events representing an activation of the secondary switch of the pointing device. (*See* http://developer.android.com/intl/zh-TW/reference/android/view/GestureDetector.OnGestureListener.html.) By way of example, this nested class includes *onLongPress()*,which generates an event representative of a right-click on a mouse after a "long press" gesture. (*See id.*) With reference to an example, Figure 14-2 shows a list of contacts displayed on the Droid 2.

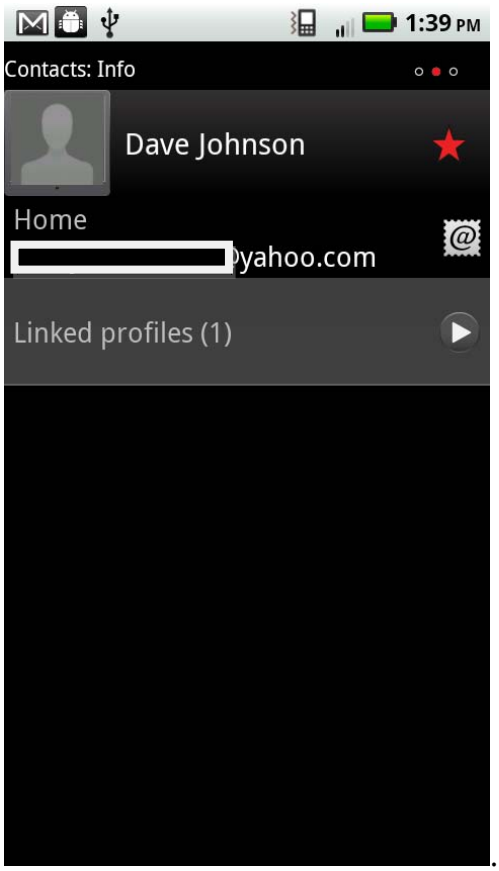
U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 14-2. When a user performs a long press on one of the contact names in the list, the application generates a context menu that overlays the contacts list view as shown in figure 14-3.

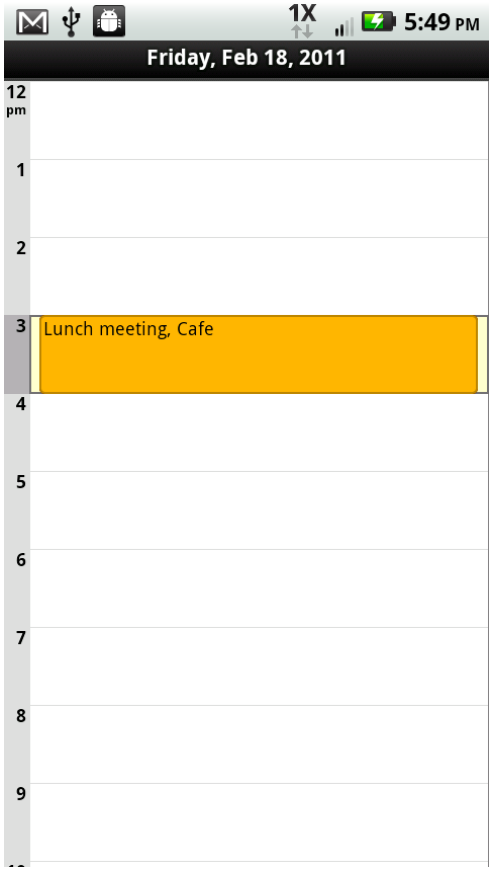
U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 14-3. Thus, the contacts list generates a context menu representing an activation of the secondary switch of a pointing device. (*See* http://developer.android.com/intl/zh-TW/guide/practices/ui_guidelines/menu_design.html) (stating that "[a] Context menu is similar to a right-click context menu in a desktop operating system. …. A user can touch & hold on content on the screen to access a Context menu.") With reference to another example, Figure 14-4 shows that Droid 2 includes a Calendar application.

U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 14-4. Figure 14-5 shows a screenshot from the Calendar application.

U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 14-5. When a user performs a long press on one of the objects representing a day, the application generates a context menu that overlays the month view, as shown in Figure 14-6.

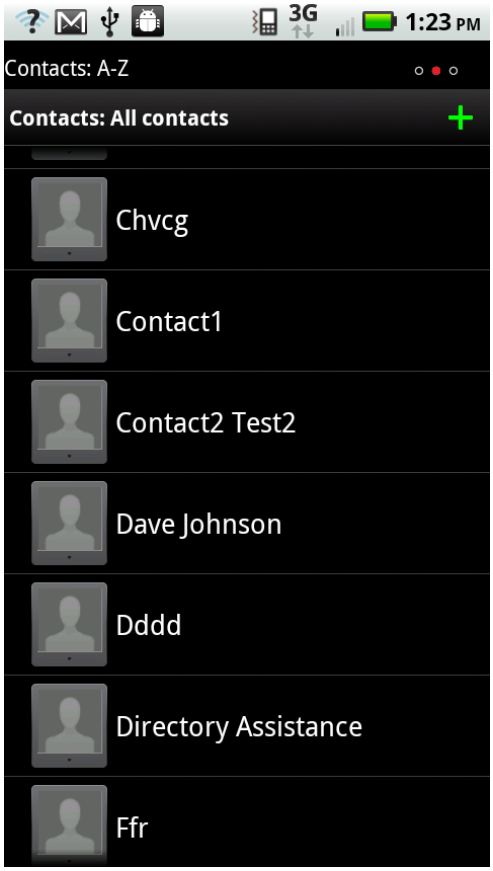
U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 14-6. Thus, the Calendar application generates a context menu representing an activation of the secondary switch of a pointing device. (*See* http://developer.android.com/intl/zh-TW/guide/practices/ui_guidelines/menu_design.html) (stating that "[a] Context menu is similar to a right-click context menu in a desktop operating system. …. A user can touch & hold on content on the screen to access a Context menu.")
in response to the stylus being removed from the touch-sensitive display surface before the threshold amount of time generating at least one event	**Each Accused Device, in response to the stylus being removed from the touch-sensitive display surface before the threshold amount of time, generates at least one event representing an activation of the primary switch of the pointing device**. *GestureDetector* determines when a user has performed a "tap"

U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
representing an activation of the primary switch of the pointing device.	gesture. The nested class *GestureDetector.OnGestureListener* then generates events representing an activation of the primary switch of the pointing device. (*See* http://developer.android.com/intl/zh-TW/reference/android/view/GestureDetector.OnGestureListener.html.) By way of example, this nested class includes *onSingleTapUp(),*which will generate an event representative of a left-click on a mouse after a "tap" gesture. (*See id.*) By way of example, in the contacts list as shown in Figure 14-2 above, when a user performs a tap on one of the contacts in the list, the application switches to a screen showing information relating to the selected contact as shown in figure 14-7  . Figure 14-7 Each Accused Device does not generate a context menu when the user performs a tap on a contact in the list. This behavior is representative of the primary switch of a pointing device – e.g., a

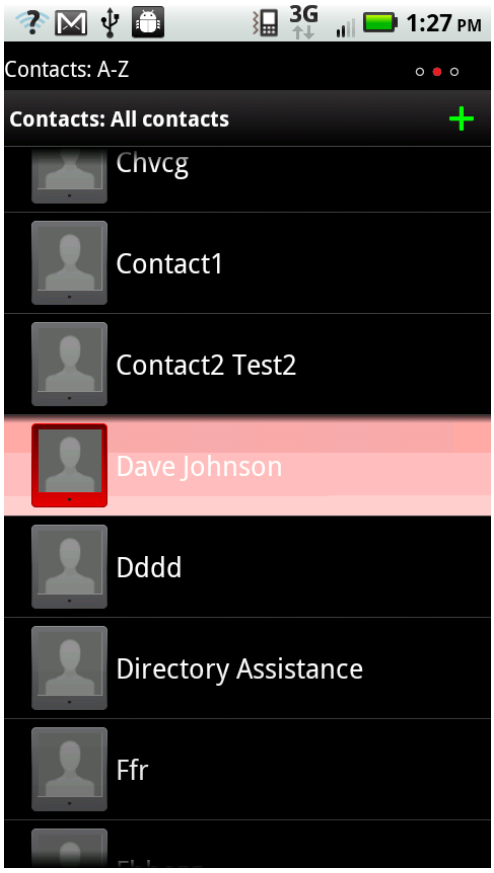
U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	"left click" on a mouse. By way of example, in the calendar application as shown in Figure 14-5 above, when a user performs a tap on one of the objects representing a day, the application switches to a screen that shows the events scheduled on the calendar for that day as shown in Figure 14-8.  Figure 14-8. The Calendar application does not generate a context menu when the user performs a tap on an object in the month view. This behavior is representative of the primary switch of a pointing device – e.g., a "left click" on a mouse.

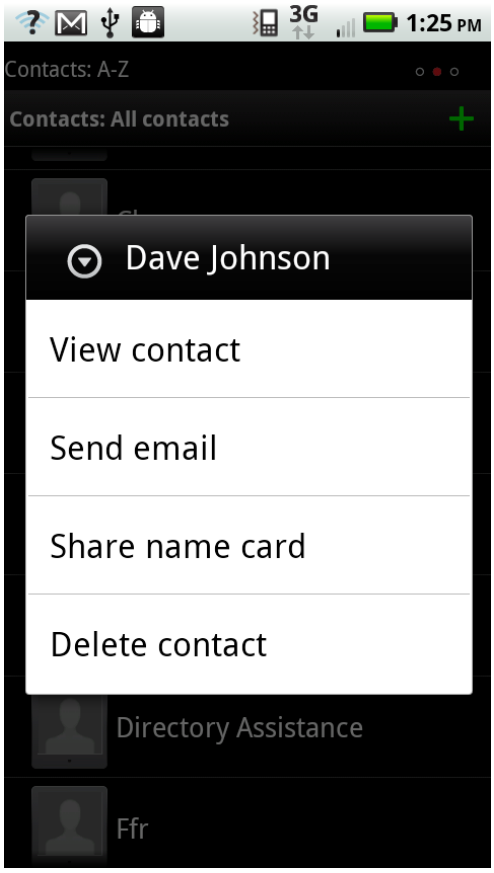
U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
16. The method of claim 14, wherein the pointing device comprises a mouse, the primary switch comprises a left button of the mouse, and the secondary switch comprises a right button of the mouse.	**Each Accused Device is configured to simulate at least one gesture of a pointing where the pointing device comprises a mouse.** Each Accused Device is configured to simulate a mouse. Such a mouse may have a "left" mouse button that serves as a primary switch and a "right" mouse button that serves as a secondary switch. (*See* http://developer.android.com/guide/practices/ui_guidelines/menu_design.html) (stating that "[a] Context menu is similar to a right-click context menu in a desktop operating system. …. A user can touch & hold on content on the screen to access a Context menu.")
17. The method of claim 14, wherein the pointing device comprises a trackball, the primary switch comprises a left button of the trackball, and the secondary switch comprises a right button of the trackball.	**Each Accused Device is configured to simulate at least one gesture of a pointing device where the pointing device comprises a trackball.** Each Accused Device is configured to simulate a trackball. Such a trackball may have a "left" button that serves as a primary switch and a "right" button that serves as a secondary switch. (*See* http://developer.android.com/guide/practices/ui_guidelines/menu_design.html) (stating that "[a] Context menu is similar to a right-click context menu in a desktop operating system. …. A user can touch & hold on content on the screen to access a Context menu.")
37. In a computer, a method for providing feedback responsive to use of a stylus on a touch-sensitive display surface, the method comprising the steps of:	**Each Accused Device is a computer configured to perform a method for providing feedback responsive to use of a stylus on a touch-sensitive display surface**. Each Accused Device is a handheld computer. (*See* http://www.motorola.com/Consumers/US-EN/Consumer-Product-and-Services/Mobile-Phones/ci.Motorola-DROID-2-US-EN.alt) (describing Motorola's computer processor and other specifications.) Each Accused Device includes the Android operating system platform ("Android"). (*See id.*) By way of example, Figure 37-1 shows the Motorola device branded as Droid 2 (hereinafter, "Droid 2").

U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 37-1. Each Accused Device is configured to provide feedback response to use of a stylus on a touch-sensitive display surface **Each Accused Device has a touch-sensitive display surface**. Each Accused Device includes a touch-sensitive display. (*See id.*) (describing the touch screen interface of the Droid 2)
detecting whether a stylus is being held down on a touch-sensitive display surface for at least a threshold amount of time; and	**Each Accused Device detects whether the stylus is held against the touch-sensitive display surface for at least a threshold amount of time**. Each Accused Device includes a processor. (*See id.*) (describing the Droid 2's computer processor and other specifications.) The processor is configured to detect whether a user presses his or her finger against the touch screen display for at least a threshold amount of time. The Android operating system on Each Accused Device provides a *GestureDetector* class for identifying "gestures." (http://developer.android.com/intl/zh-TW/reference/android/view/GestureDetector.html.) *GestureDetector* analyzes user input events and determines which gestures the user has performed. In one exemplary usage, *GestureDetector* detects "tap" and "long press" gestures. For a "tap" gesture, the user's finger is held against the touch screen for less than a threshold amount of time and for a "long press" gesture, the user's finger is held against the touch screen for at least a threshold amount of time. The threshold is identified as a

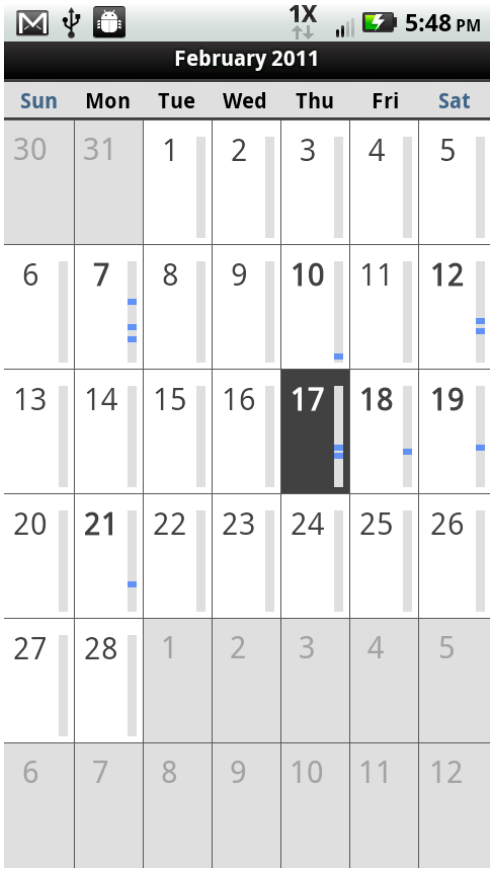
U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	constant in the *GestureDetector* class. *(See android/frameworks/base/core/java/android/view/GestureDetector.java)*
generating a state change indicator responsive to the stylus being held down for at least the threshold amount of time.	**<u>Each Accused Device generates a state change indicator</u>**. With reference to an example, Figure 37-2 shows the Droid 2 displaying a list of contact entries.  Figure 37-2. The user may perform a long press on a contact name in the list. When a user places a finger down on a contact name in the list, the device highlights the background of the contact name in a solid color, such as red, as shown in Figure 37-3.

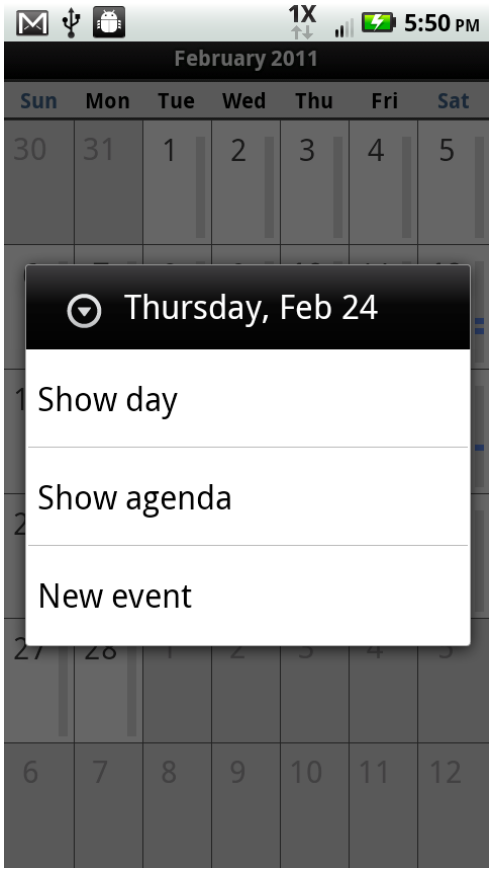
U.S. Patent No. 6,791,536 (‘536 Patent)	Accused Devices
	 Figure 37-3. While the user keeps the finger down on the touchscreen, the color of the background of the selected contact name fades to a lighter shade as shown in Figure 37-4.

U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 37-4. After a delay, the selected contact's background changes back to the original background color. After the color changes back to the original background color, the device displays a context menu as shown in Figure 37-5.

U.S. Patent No. 6,791,536 (‘536 Patent)	Accused Devices
	 Figure 37-5 In this example, the sequence of color changes of the contact name background is a state change indicator. With reference to another example, Figure 37-6 shows that Droid 2 includes a Calendar application.

U.S. Patent No. 6,791,536 (‘536 Patent)	Accused Devices
	 Figure 37-6. Figure 37-7 shows a screenshot from the Calendar application.

U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 37-7. A user may perform a long press on a day in the calendar. When a user places a finger down on a day in the calendar, the device highlights the day by filling the day in a solid color, such as grey, as shown in Figure 37-7. After a delay, the device changes the fill color of the day to the original color. After the color of the day changes, the device displays a context menu as shown in Figure 37-8.

U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 37-8. In this example, the changing color of the selected day is the state change indicator. With reference to yet another example, when a user performs a long press on an icon on the home screen such as the browser icon shown in figure 37-9, the device provides a state change indicator.

U.S. Patent No. 6,791,536 (‘536 Patent)	Accused Devices
	 Figure 37-9. In response to a long click of an icon on the home screen, the device generates tactile feedback in the form of a vibration impulse and then enters a mode where the user can move home screen icons. As shown in figure 37-10.

U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	 Figure 37-10 In this example, the vibration impulse is a state change indicator.
38. The method of claim 37, wherein the step of generating includes generating a visual state change indicator on the touch-sensitive display surface.	**<u>Each Accused Device generates a visual state change indicator on the touch-sensitive display surface.</u>** In the example described above in relation to the contacts list as shown in figure 37-2 – 37-5, the device causes the color of the background of a contact name to fade from red to a lighter shade and then change back to its original color. In this way the device generates a visual state change indicator on the touch-sensitive display surface In the example described above in relation to the calendar application shown in Figures 37-6 – 37-8, the device changes the

U.S. Patent No. 6,791,536 ('536 Patent)	Accused Devices
	color of the selected day before displaying a context menu. In this way, the device generates a visual state change indicator on the touch-sensitive display surface.
39. The method of claim 37, wherein the step of generating includes generating a visual state change indicator at a location on the touch-sensitive display surface depending upon a location of the stylus.	**Each Accused Device generates a visual state change indicator at a location dependent upon the stylus.** In the example described above in relation to the contacts list as shown in Figures 37-2 – 37-5, the device causes the color of the background of a contact name to fade from red to a lighter shade of red and then change back to its original color. Because the device changes the color of the contact name background that the user selects with a finger, the visual state change indicator location depends upon the location of the stylus. In another example described above in relation to the calendar application shown in Figures 37-6 – 37-8, the device changes the color of the selected day before displaying a context menu. Because the device changes the color of the day that the user selects with a finger, the visual state change indicator location depends upon the location of the stylus.
40. The method of claim 37, wherein the step of generating includes generating an animated visual state change indicator on the touch-sensitive display surface.	**Each Accused Device generates an animated visual state change indicator.** In the example described above in relation to the contacts list as shown in Figures 37-2 – 37-5, the device causes the color of the background of a selected contact name to fade from red to a lighter shade of red and then change back to its original color. Because the state change indicator changes over time, the Droid 2 generates an animated state change indicator.