

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.,

Petitioner

v.

SMITH INTERFACE TECHNOLOGIES, LLC

Patent Owner

IPR2024-01084

Patent No. 10,649,578

DECLARATION OF DR. VIJAY MADISETTI

TABLE OF CONTENTS

	Page
I. INTRODUCTION.....	4
II. PROFESSIONAL BACKGROUND.....	4
III. COMPENSATION.....	10
IV. LEGAL CONSIDERATIONS.....	10
A. Anticipation.....	11
B. Obviousness	11
C. Claim Interpretation	12
V. OVERVIEW OF TASK & BASIS FOR OPINIONS.....	13
VI. OVERVIEW OF THE '578 PATENT.....	14
A. Pattern-Based Gestures.....	15
B. Independent Claims 1, 15, and 27	21
C. Dependent Claims 2-3, 16-17, and 28-29.....	25
VII. LEVEL OF ORDINARY SKILL IN THE ART	26
VIII. CLAIM CONSTRUCTION.....	26
A. Conditional Limitations.....	26
1. Claims 1 and 27.....	26
2. Claim 15.....	31
B. “The Second Input”	33
IX. OVERVIEW OF THE PRIOR ART	35
A. Shiplacoff.....	35
B. Nan	40
X. THE PETITION’S GROUND 1 FAILS TO ESTABLISH THAT THE PRIOR ART DISCLOSES THE CHALLENGED CLAIMS.....	41
A. Limitations [1i]-[1k], [15e]-[15g], [27d]-[27f]: Shiplacoff Fails to Disclose or Suggest Programming Logic Capable of Triggering the Recited Display Actions in the Required Operational Context	41

B.	Limitations [1j]-[1k], [15f]-[15g], [27e]-[27f]: Shiplacoff Does Not Disclose or Suggest Using a “Movement Threshold” To Differentiate Between Inputs.....	48
C.	Limitations [1k], [15g], [27f]: The Petition’s Mapping of Shiplacoff’s “Tap or Flick” Input Fails for Several Reasons	52
1.	Shiplacoff Does Not Disclose or Suggest That Its “Tap or Flick” Input Triggers the Recited Display Action.....	52
2.	A POSITA Would Not Have Been Motivated To Modify Shiplacoff Such That Its “Tap or Flick” Input Triggers “Replacing the User Interface of the Application with the Home Screen”	57
3.	Shiplacoff’s “Tap” Input Does Not Meet the Requirements of the Claimed “Second Input”	66
XI.	THE PETITION’S GROUND 2 FAILS TO DEMONSTRATE THAT A POSITA WOULD HAVE BEEN MOTIVATED TO COMBINE SHIPLACOFF AND NAN AS PROPOSED.....	68
XII.	THE PETITION FAILS TO ESTABLISH THAT SHIPLACOFF’S “TAP OR FLICK” INPUT DISCLOSES DEPENDENT CLAIMS 2-3, 16-17, AND 28-29.....	71
XIII.	CONCLUSION	72

I. INTRODUCTION

1. My name is Dr. Vijay Madiseti. I have been retained by Smith Interface Technologies, LLC as an independent expert consultant in this *inter partes* review (“IPR”) proceeding before the United States Patent and Trademark Office (“PTO”) regarding US Patent No. 10,649,578 (“the ’578 Patent”).

II. PROFESSIONAL BACKGROUND

2. All of my opinions stated in this Declaration are based on my own personal knowledge and professional judgment. In forming my opinions, I have relied on my knowledge and experience in designing, developing, and researching the technology referenced in this Declaration.

3. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth herein. I understand that a copy of my current curriculum vitae, which details my education and professional and academic experience, is being submitted as Exhibit 2005. The following provides a brief overview of some of my experience that is relevant to the matters set forth in this Declaration.

4. I received my Bachelor of Technology (Honors) in Electronics and Electrical Communication Engineering at the Indian Institute of Technology (IIT) in Kharagpur, India in 1984. I obtained my Ph.D. in Electrical Engineering and Computer Science at the University of California at Berkeley in 1989. I received

the Demetri Angelakos Outstanding Graduate Student Award from the University of California at Berkeley and the IEEE/ACM Ira M. Kay Memorial Paper Prize in 1989. I am now a tenured Full Professor in the Colleges of Computing and Engineering at Georgia Tech and serve as its representative to the European Telecommunications Standard Institute (ETSI). I am knowledgeable and familiar with topics including software engineering, web applications, cloud computing, user interface design, wireless networks, and associated software and firmware design—including software and firmware design for wireless and telecommunications terminals and base stations in general and ETSI/3GPP/3GPP2 standards based cellular architecture and infrastructure. I have created and taught undergraduate and graduate courses in hardware and software design for computer science and engineering applications at Georgia Tech for the past thirty years. I have been the principal investigator (“PI”) or co-PI in several active research programs in these areas, including DARPA’s Rapid Prototyping of Application Specific Signal Processors, the State of Georgia’s Yamacraw Initiative, the United States Army’s Federated Sensors Laboratory Program, and the United States Air Force Electronics Parts Obsolescence Initiative. I have received an IBM Faculty Award and NSF’s Research Initiation Award.

5. I have designed several specialized computer and communication systems over the past two decades at Georgia Tech for tasks such as wireless

audio/video processing and messaging/protocol processing for portable platforms, such as cell phones and Personal Digital Assistants. I also developed UIs for tools for electronic design automation of complex chipsets and their reverse engineering processes for the U.S. Air Force and Lockheed Martin and Boeing in the late 1990s and early 2000s. I have worked on designing systems that are efficient from performance, size, weight, area, and thermal considerations. I have developed courses and classes for the industry on these topics, and many of my lectures in advanced computer system design were developed under the sponsorship of the United States Department of Defense in the late 1990s and have been used by several U.S. and international universities as part of their course work. I graduated more than 20 Ph.D. students that now work as professors or in technical positions around the world.

6. Over the years, I have been an active consultant to industry and various research laboratories (including Massachusetts Institute of Technology Lincoln Labs and Johns Hopkins University Applied Physics Laboratory). As part of that involvement in the industry, I have been actively involved in product development. For example, in the early 2000s, I designed three GSM multiband mobile phones for a leading telecom equipment manufacturer in Asia, and in the mid-2000s, I developed wireless baseband and protocol stack software and assembly code for a leading telecommunications handset vendor that focused on efficient realization of

speech codecs and echo-cancellation and for another in the optimization of their 3G software stack. My work in this regard included creation of software code, and analysis and revision of existing software code.

7. I have also founded three companies in the areas of embedded software, military chipsets involving imaging technology, and wireless communications. The first of these companies, VP Technologies, offers computing systems for helicopter imaging systems for the United States Air Force. The second of these companies, Soft Networks, LLC, offers software for multimedia and wireless computing platforms, including the development of a set-top box for Intel that decodes MPEG-2 video streams and imaging codes for multimedia phones. The third of these companies, Elastic Video, uses region-of-interest based video encoding to capture high quality video at low bit rates, with primary application for wireless video systems. I also have extensive experience developing user interfaces for products. Recent examples of such work include my work developing a crypto exchange prototype, and my work on a web-based document management environment, called CatchUp, which is now commercialized as CatchUp Cloud (<https://catchup.cloud>).

8. I have authored more than 110 refereed journal publications and around forty peer-reviewed conference publications. I have served as associate editor or on the editorial board for technical journals, including IEEE Transactions on Circuits & Systems II, International Journal in Computer Simulation, and International

Journal in VLSI Signal Processing. I have also authored or co-authored several books, including VLSI Digital Signal Processors (IEEE Press 1995) and the Digital Signal Processing Handbook (CRC Press, 1998, 2010). I also co-authored Quick-Turnaround ASIC Design in VHDL (Kluwer Academic Press 1996) and Platform-Centric Approach to System-on-Chip (SoC) Design (Springer 2004). I am also the editor of several books, including the three-volume DSP Handbook set: Volume 1: Digital Signal Processing Fundamentals, Volume 2: Video, Speech, and Audio Signal Processing and Associated Standards, and Volume 3: Wireless, Networking, Radar, Sensory Array Processing, and Nonlinear Signal Processing, published in 2010 by CRC Press, Boca Raton, Florida. More recently I have authored Cloud Computing (2014, CreateSpace Press), and Internet of Things (2014, CreateSpace), and the book, Cloud Computing, was nominated as a Notable Book of 2014 by the Association of Computing Machinery (ACM) in July 2015.

9. I have worked in the area of user interfaces over the past twenty years as part of my extensive development in mobile and portable applications for mobile phones and portable devices since the early 2000 timeframe. I developed commercial products for leading companies, and was part of the team that developed the user interfaces for a Soft Switch, called SNX 8500, that provides voice and VOIP services for enterprises. In addition to authoring textbooks and research papers in the area of wireless messaging interfaces, I have developed web-based document

management systems and web-based fintech applications. I have submitted approximately 100 invention disclosures and provisional patents over the past ten years. I am listed as a first inventor on over thirty issued US patents. Many of my patents disclose techniques for efficient user interface design including use of multilevel hierarchies and other UI innovations in the area of user interfaces for work environments.

10. I have been extensively involved in the activities of one of the premier standard setting organizations in the world, the IEEE, since the 1980s, and was a member of the IEEE Press Board from 1995-1997. I was the Technical Program Chair of IEEE ICASSP in 1996. I was also the Associate Editor of IEEE Transactions on Circuits & Systems II from 1994 to 1997. I am also the director of the IEEE Atlanta Chapter, and the author of several drafts of IETF proposals in the area of VOIP and Voice/Video streaming over the internet. I have been elected a Fellow of the IEEE. The Fellow is the highest grade of membership of the IEEE, a world professional body consisting of over 300,000 electrical and electronics engineers, with only one tenth of one percent (0.1%) of the IEEE membership being elected to the Fellow grade each year. Election to Fellow is based upon votes cast by existing Fellows in IEEE.

11. I have also been awarded the 2006 Frederick Emmons Terman Medal by the American Society of Engineering Education for contributions to Electrical

Engineering, including authoring a widely used textbook in the design of VLSI digital signal processors. I was awarded VHDL International Best Ph.D. Dissertation Advisor Award in 1997 and the NSF RI Award in 1990. I was Technical Program Chair for both the IEEE MASCOTS in 1994 and the IEEE Workshop on Parallel and Distributed Simulation in 1990.

III. COMPENSATION

12. My work on this matter is being billed at my customary rate of \$700 an hour. Also, I am being reimbursed for reasonable expenses I incur in relation to my services. I have no pecuniary interest in the outcome of this proceeding. I understand I will be paid regardless of the outcome of any proceeding in which my work is used.

IV. LEGAL CONSIDERATIONS

13. I am not an attorney and have not been asked to offer my opinion on the law. However, as an expert offering an opinion on whether the claims of the '578 Patent are patentable, I have been told that I am obliged to follow existing law. My understanding of the law is based on information provided by counsel.

14. I understand that in an IPR proceeding, a patent claim is unpatentable if it is shown by a preponderance of the evidence that the claim is anticipated or obvious. I also understand that the anticipation or obviousness determination is viewed from the perspective of a person of ordinary skill in the art ("POSITA") at

the priority date of the '578 Patent, which I have been asked to treat as August 5, 2011.

A. Anticipation

15. I understand that a reference anticipates a claim if it discloses each and every element recited in the claim, arranged as in the claim, so as to enable a POSITA to make and use the claimed invention without the need for undue experimentation in light of the general knowledge available in the art. I understand that a U.S. Patent document is presumed to have sufficient description to include sufficient detail for a POSITA to make and use the subject matter that document describes.

16. I also understand that a claim limitation may be present in an alleged prior art reference either expressly or inherently, but inherency may be established only if such limitation is necessarily present in the reference. I understand inherency cannot be established by probabilities or possibilities, and the mere fact that a certain thing may result from a given set of circumstances is not sufficient to establish inherency.

B. Obviousness

17. For a claim to be obvious, I understand that every limitation of the claim must be disclosed in a combination of prior art references, as viewed from the perspective of a POSITA. I understand that a POSITA is presumed to know the

relevant prior art.

18. I understand that while there is no rigid rule that a reference or combination of references must contain a “teaching, suggestion, or motivation” to combine references, some motivation to combine or modify prior art references must exist in the references or the knowledge generally available to a POSITA, otherwise the challenged claims would not have been obvious. I also understand that the “teaching, suggestion, or motivation” test can be a useful guide in establishing a rationale for combining elements of the prior art. In addition, I have been informed and I understand that there must be a reasonable expectation of success in combining prior art disclosures to arrive at the claimed invention.

C. Claim Interpretation

19. I understand that a claim term is interpreted according to its ordinary and customary meaning as a POSITA would have understood the term in light of the surrounding claim language, other claims, the specification, and the patent’s prosecution history, which are referred to as intrinsic evidence. I also understand that prior art references cited in the patent’s prosecution history are considered intrinsic evidence. I further understand that evidence outside the patent and its prosecution history (e.g., dictionaries and technical articles, as well as the POSITA’s general knowledge and experience) may inform the context in which a POSITA would have understood the claims of a patent. I understand this ordinary and

customary meaning applies absent unique circumstances, such as where a patent clearly expresses an intent to set forth a special meaning for a term.

V. OVERVIEW OF TASK & BASIS FOR OPINIONS

20. I have been asked to review the '578 Patent. I have been asked to provide opinions related to certain issues from the perspective of a POSITA, having knowledge of the relevant art, as of the priority date, and—except where otherwise noted—the opinions stated in this declaration are from that perspective. The qualifications and abilities of such a person are described in Section VII below.

21. More particularly, I have been informed that the Patent Trial and Appeal Board (the “Board”) will decide whether the Petition (Paper 2) (“Pet.”) establishes by a preponderance of the evidence that at least one claim of the '578 Patent is unpatentable. I have been asked to focus my analysis on the Petition’s proposed invalidity theories related to the challenged claims of the '578 Patent.

22. In preparing this declaration, I have considered the '578 patent in its entirety and the general knowledge of a POSITA as of the priority date. My opinions are based on my education, training, and experience, as well as items that I reviewed to prepare my opinions, including the Petition, the Board’s Institution Decision (Paper 6) (“ID”), and the items listed in the following table:

Exhibit	Shorthand	Description
1001	'578 Patent	U.S. Patent No. 10,649,578
1002	'578 Patent FH	U.S. Patent No. 10,649,578 File History

1003	Terveen	Declaration of Dr. Loren Terveen
1004	Shiplacoff	U.S. Pat. App. Pub. No. 2010/0095240
1005	Nan	U.S. Pat. No. 9,658,766
1006	N/A	First Complaint, <i>Smith Interface Technologies, LLC v. Apple Inc.</i> , 3-23-cv-01187, S.D. Cal. (June 27, 2023)
1007	N/A	Exhibit A2 of Infringement Contentions, <i>Smith Interface Technologies, LLC v. Apple Inc.</i> , 3-23-cv-01187, S.D. Cal. (June 5, 2024)
1008	Shaffer	U.S. Pat. No. 9,311,112
1009	Duarte	U.S. Pat. No. 8,547,244
1010	Matthews	U.S. Pat. No. 8,375,336
1011	Shiplacoff '989	U.S. Pat. App. Pub. No. 2010/0185989
1012	Jacobs	U.S. Pat. No. 9,716,874
2003	Duarte-992	U.S. Pat. App. No. 12/115,992
2006	Terveen Deposition	Deposition of Loren Terveen, Ph.D. (April 25, 2025)
2007	'918 Provisional	U.S. Prov. Pat. App. No. 61/581,918
2008	Anzures	U.S. Pat. App. Pub. No. 2009/0174680
2009	Jobs	U.S. Pat. App. Pub. No. 2008/0122796

I understand that other issues may arise that require further explanation, and I will provide that explanation if appropriate. As a result, I respectfully reserve the right to update and supplement this declaration and the information and opinions provided herein.

VI. OVERVIEW OF THE '578 PATENT

23. Based on my experience, at the time of the '578 Patent's priority date,

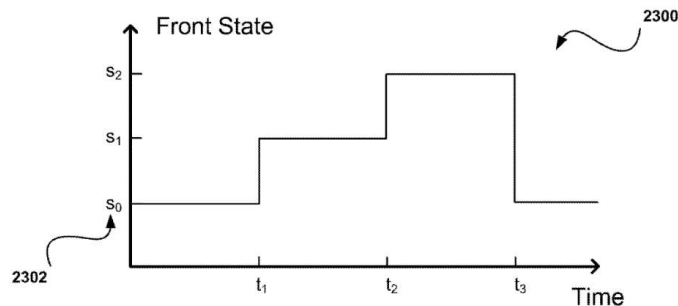
touchscreen mobile devices were becoming increasingly capable of more complex tasks that required more control than possible with basic input methods. Further, the limited screen space of such devices meant that the input and output techniques used for more complex tasks on laptop and desktop computers could not be used. The '578 Patent addresses these shortcomings by enabling users to operate these new, powerful mobile devices and their feature-rich applications through implementation of novel “pattern-based gestures,” which are gestures defined by different attributes and contexts that enable users to trigger a variety of corresponding actions.

A. Pattern-Based Gestures

24. The '578 Patent describes “pattern-based gestures” as “an ordered sequence” of one or more “gesture attributes,” also referred to as “touch event attributes,” that may be recognized using a “decision tree.” '578 Patent, 74:33-35, 74:14-16, 54:10-16; '918 Provisional, ¶¶[00579], [00581], [00359]. These gestures are described in the '578 Patent as “a path through a phase space of one or more dimensions.” '578 Patent, 72:36-38; '918 Provisional, ¶[00564]; *see also* '578 Patent, 75:3-10 (“[P]robabilistic pattern recognition algorithms may output a probability... [where] this probability may be used to alter or modify a selection.”); '918 Provisional, ¶[00585].

25. As an example, multi-part “pattern-based gesture 2302” is differentiated from other gestures by a unique “phase space path which represents

the time evolution of the touch state associated with a contact point” on the touchscreen:



’578 Patent, 72:36-55, FIG. 23; ’918 Provisional, ¶[00566], FIG. 28. As shown, a gesture’s phase space may include one or more “touch state transitions,” each defining attributes such as “contact point changes” and “the time spent on each leg of the path through phase space.” ’578 Patent, 72:50-73:16; ’918 Provisional, ¶¶[00566]-[00570]. “[P]attern-based gesture 2302” includes three touch state transitions: a first transition in which, “[a]t time t1, the contact point changes from touch state s0 to s1,” as well as a second and third transition in which “[s]imilar changes of state occur at t2 and t3,” respectively. ’578 Patent, 72:56-64; ’918 Provisional, ¶[00567].

26. Different gestures may be defined by different touch states, transitions, time spent at each touch state and transition, and other factors. ’578 Patent, 72:65-73:16; ’918 Provisional, ¶[00568]; *see also, e.g.*, ’578 Patent, 73:37-62 (describing other exemplary pattern-based gestures 2304 and 2306); ’918 Provisional, ¶[00573]-[00577]. Multiple pattern-based gestures may share the same starting points (e.g.,

s0) and even intermediate points (e.g., s1), but each individual gesture can nonetheless be differentiated using decision-tree logic. '578 Patent, 74:33-35; '918 Provisional, ¶[00581]. The '578 Patent describes using “pattern recognition algorithms and methods” to analyze the attributes of a user input to determine whether it meets all the attributes for a defined gesture. '578 Patent, 74:43-75:15, 65:31-39; '918 Provisional, ¶¶[00582]-[00586], [00444]; *see also, e.g.*, '578 Patent, 18:1-10 (explaining that “initial contact pressure” may also be used to differentiate an “accidental” input); '918 Provisional, ¶[00101].

27. The '918 Provisional, which the '578 Patent incorporates by reference ('578 Patent, 1:8-24), also discloses techniques for “gesture differentiation,” which is distinguished from simply detecting a “basic user interaction (i.e., button press, basic application interaction, etc.).” '918 Provisional, ¶¶[00498]-[00502], FIG. 24. As shown in the '918 Provisional's FIG 24 (which I have reproduced below), the process begins by “determin[ing] whether there is any meaningful user input,” including by applying a “threshold above which user interaction may rise before it may be deemed meaningful.” '918 Provisional, ¶¶[00501]-[00502]. Thereafter, the attributes of the user input are analyzed to determine whether they correspond a particular gesture. *Id.*, ¶¶[00503]-[00529], FIG. 24.

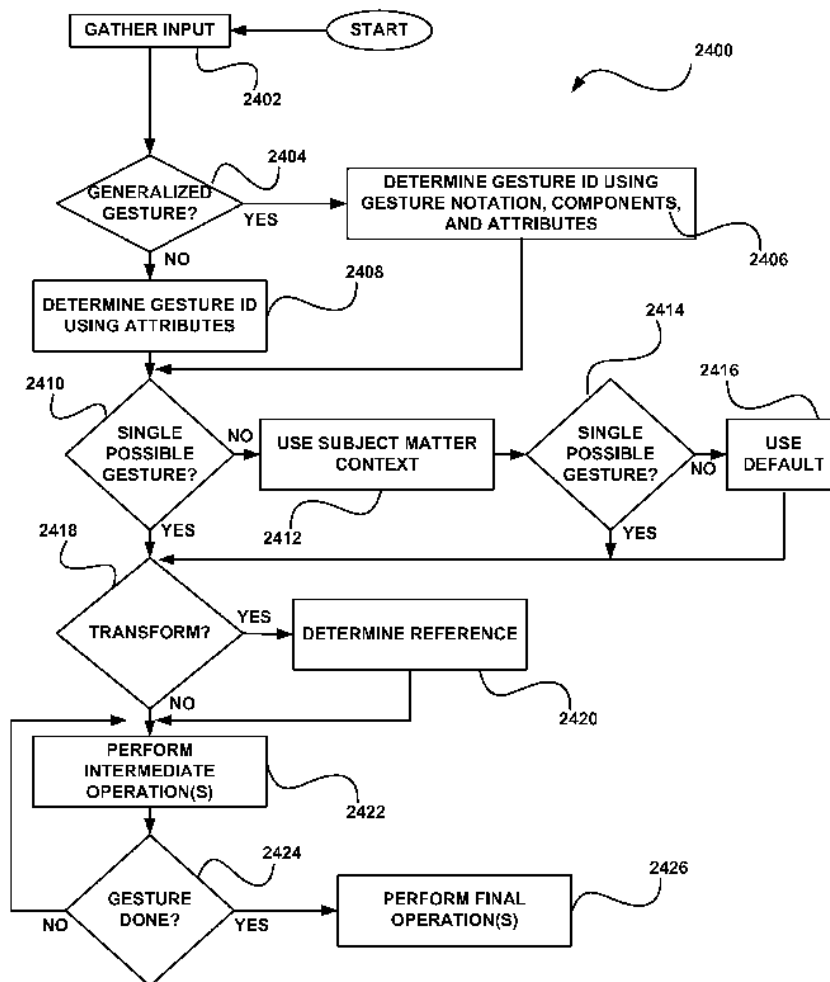


FIGURE 24

28. Further, as shown in the '918 Provisional's FIG. 24, determining whether a user input corresponds to a defined gesture may also depend on the "gesture subject matter context." *Id.*, ¶[00512]. Specifically, in addition to attributes, gestures may "be distinguished from each other through the context of their use." '578 Patent, 65:20-22; '918 Provisional, ¶[00444]. For instance, the '578 Patent explains that a user input with the same or similar touch state attributes may trigger a different operation depend on "application context," e.g., "which

application is active” during the user’s contact with the touchscreen. ’578 Patent, 60:12-17 (explaining that “a gesture which scrolls a text field in one application may turn a page when performed in a different application”); ’918 Provisional, ¶[00404]. As another example, the ’918 Provisional explains that “gesture subject matter context” may include whether a certain object is being displayed and/or has been selected, as well as the location of the object relative to the user’s contact with the touchscreen. ’918 Provisional, ¶[00512]-[00513]. Thus, a gesture may be defined to require a starting point that is on or near a specific object displayed on the touchscreen. *Id.*

29. The ’578 Patent further explains that each gesture may be mapped to a particular command or operation, such as “scrolling a display or user interface element, zooming in or out, modifying the display brightness, and/or... placing the system in a selection mode.” ’578 Patent, 55:1-16, Limitations [1h], [1j]-[1k]; ’918 Provisional, ¶[00369]-[00370]. Thus, for instance, “the association between gesture and operation may be context-dependent.” ’578 Patent, 60:11-12, Limitations [1i]-[1k]; ’918 Provisional, ¶[00404].

30. The ’578 Patent’s FIG. 16 (and related portions of the specification) illustrates the type of decision-tree-based programming logic used to determine whether a user’s contact with the touchscreen corresponds to a pre-defined gesture and, if so, trigger the action mapped to the particular gesture that is detected:

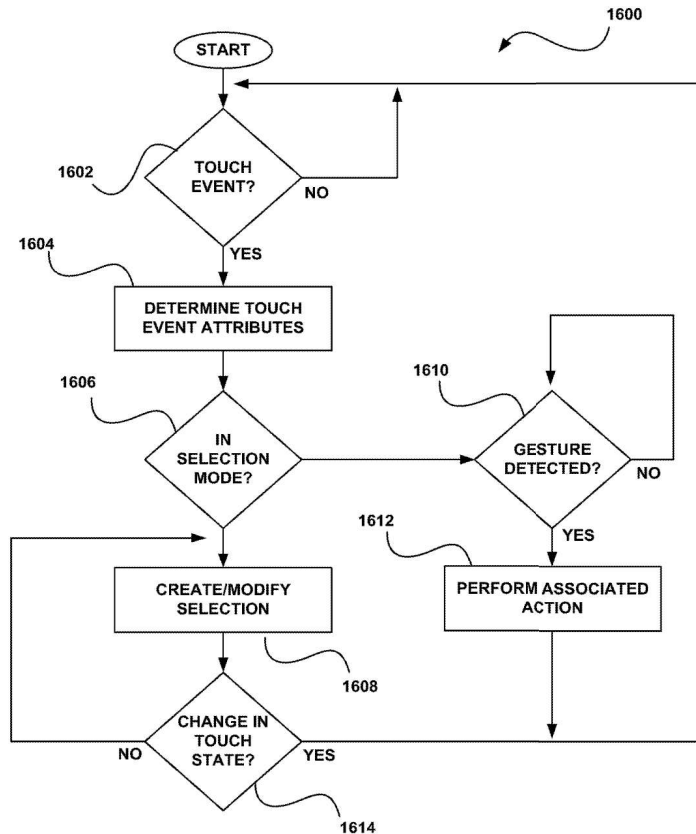


FIGURE 16

'578 Patent, FIG. 16, 53:57-55:16; '918 Provisional, FIG. 16, ¶¶[00355]-[00370]. The particular logic illustrated in FIG. 16 enables the device to determine whether a user's contact with the touchscreen corresponds with "a selection made within a user interface." '578 Patent, 53:57-65; '918 Provisional, ¶[00355]. As illustrated, the logic operation begins with detection of a touch event and determination of the "touch event attributes" for that event. '578 Patent, 53:57-54:9, Limitations [1i]-[1k]; '918 Provisional, ¶¶[00355]-[00357]. "Touch event attributes" can include any measurable aspect of the user interaction with the touchscreen, such as touch state, contact area, contact point, and touch event attributes over time. '578 Patent, 54:10-

16; '918 Provisional, ¶[00359]; *see also* '578 Patent, 56:45-16 (“Possible gesture inputs may include, but are not limited to, touch event attributes (both initial and over time)[.]”); '918 Provisional, ¶[00386].

31. After determining the attributes of the touch event that started the logic operation, a determination is made, at step 1606, as to whether the system is in a “selection mode.” '578 Patent, 54:17-18; '918 Provisional, ¶[00360]; *see also* '578 Patent, 76:19-24 (describing “selection mode” as another exemplary “context”); '918 Provisional, ¶[00593]. “If it is determined in 1606 that the system is not in a selection mode,” then a determination must be made, at step 1610, whether the touch attributes match a predefined gesture (i.e., whether “a gesture has been detected”). '578 Patent, 54:61-63, Limitations [1j]-[1k]; '918 Provisional, ¶[00368]. “If a gesture is detected, then an associated action is performed” (step 1612). '578 Patent, 55:1-2, Limitations [1j]-[1k]; '918 Provisional, ¶[00369]. On the other hand, “[i]f the system is in a selection mode,” the same touch event attributes may instead trigger a “selection” operation such that “a selection is created and/or modified as a function of the touch event attributes [and] then displayed.” '578 Patent, 54:35-37; '918 Provisional, ¶[00363].

B. Independent Claims 1, 15, and 27

32. The Petition challenges the three independent claims of the '578 Patent: Claims 1, 15, and 27. Claim 1 is directed to an “electronic device” with “one or

more programs including instructions for” causing the device to perform certain logic operations. Among other things, Claim 1 recites “instructions for” causing the claimed device to “detect[] a second input by a second contact” that: (1) “includes movement across the display in a direction,” and (2) occurs “while [the device is] displaying the user interface of the application.” ’578 Patent, Limitation [1i]. Claim 1 further recites a logic operation that is executed “in response to detecting *the* second input” and causes the device to (a) determine whether or not the “second input” meets specified “criteria,” and (b) “in accordance with” that yes/no determination, perform a specified display action. *Id.*, Limitations [1j]-[1k]. Limitations [1j] and [1k] recite the display action to be performed by the device depending on whether the input meets the specified criteria. In particular, Limitations [1j]-[1k] recite¹:

- [1j] “in accordance with a determination that *the second input meets one or more criteria* that is met when the second input is detected to include a movement parameter that is above a movement threshold,” the instructions cause the device to “*display[] at least a portion of the user interface of the application in a first virtual display layer that*

¹ All emphasis added throughout unless otherwise noted.

appears at a lesser depth as compared to a second virtual display layer, such that the at least portion of the user interface of the application is reduced in size and is further displayed in its entirety when displayed in the first virtual display layer”; and

- [1k] “in accordance with a determination that ***the second input does not meet the one or more criteria*** that is met when the second input is detected to include the movement parameter that is above the movement threshold,” the instructions cause the device to “***replac[e] the user interface of the application with the home screen including the icon associated with the application.***”

33. Claim 15 is substantially similar to Claim 1 but directed to a method. Claim 15 recites method steps (i.e., Limitations [15b]-[15g]) that are analogous to Limitations [1f]-[1k] of Claim 1.

34. Claim 27 is directed to a “non-transitory computer readable storage medium” including “instructions” that, when executed, “cause the electronic device to” perform certain logic operations. Like Claim 1, Claim 27 recites instructions that cause the claimed device to “detect[] a second input by a second contact” that: (1) “includes movement across the display in a direction,” and (2) occurs “while [the device is] displaying the user interface of the application.” ’578 Patent, Limitation [27d]. Also like Claim 1, Claim 27 recites a logic operation that is executed “in

response to detecting the second input” and causes the device to determine whether or not the “second input” meets specified “criteria” and, based on that determination, perform a specified display action. *Id.*, Limitations [27e]-[27f]. Claim 27 differs from Claim 1 in that the recited logic operation specifies two different “criteria” against which the “second input” is evaluated. In particular, Limitations [27e]-[27f] recite:

- [27e] “in accordance with a determination [i] that ***the second input meets a first one or more criteria*** that is met when the second input is detected to include a movement parameter that is above a movement threshold and [ii] that ***the second input that meets a second one or more criteria*** that is met when the second input is detected to include a duration parameter that is above a duration threshold,” the instructions cause the device to “***display at least a portion of the user interface of the application in a first virtual display layer that appears at a lesser depth as compared to a second virtual display layer***, where the at least portion of the user interface of the application is reduced in size and is further displayed in its entirety when displayed in the first virtual display layer”; and
- [27f] “in accordance with a determination that ***the second input meets the first one or more criteria*** that is met when the second input is

detected to include the movement parameter that is above the movement threshold and that ***the second input does not meet the second one or more criteria*** that is met when the second input is detected to include the duration parameter that is above the duration threshold,” the instructions cause the device to “replace the user interface of the application with the home screen including the icon that corresponds to the application.”

C. Dependent Claims 2-3, 16-17, and 28-29

35. Dependent Claims 2-3, 16-17, and 28-29 recite additional requirements that must be met by “***the*** second input” of Limitations [1i]-[1k], [15e]-[15g], and [27d]-[27f]. *Compare* ’578 Patent, Limitations [1i]-[1k], [15e]-[15g], [27d]-[27f], *with id.*, Claims 2-3, 16-17, 28-29.

36. In particular, Dependent Claims 2, 16, and 28 recite that “the second contact” of “the second input” must include “movement that starts at an edge of the display.” Thus, in each instance of the claimed logic operation, the detected “second input” must share the same starting point: “at an edge of the display.”

37. Dependent Claims 3, 17, and 29 recite that “the second contact” of “the second input” must include “movement across the display [in] a vertical direction.” Thus, in each instance of the claimed logic operation, the “second input” must include a “vertical direction” of movement.

VII. LEVEL OF ORDINARY SKILL IN THE ART

38. I understand that a patent must be analyzed from the perspective of a POSITA as of the patent's priority date, which is typically considered to be the patent's earliest filing date.

39. I agree with the level of ordinary skill in the art adopted by the Board for the purposes of the Institution Decision. *See* ID, 5. Specifically, a POSITA:

would have had a bachelor's degree in computer science, computer engineering, electrical engineering, human-computer interaction, or a comparable field and at least two years of professional experience in the design and development of graphical user interfaces (GUIs), human-computer interfaces, or equivalents thereof.

Id.

VIII. CLAIM CONSTRUCTION

A. Conditional Limitations

1. Claims 1 and 27

40. Limitations [1i]-[1k] of Claim 1 collectively recite “one or more programs including instructions for” “detecting a second input by a second contact” that “includes movement across the display in a direction” and occurs “while [the device is] displaying the user interface of the application,” determining whether “the second input” “meets” or “does not meet” specified criteria, and “in accordance with [that] determination” performing a corresponding display action. A POSITA would have understood Limitations [1i]-[1k] to require programming logic that is executed

in a specific operational context—“while [the device is] displaying the user interface of the application”—to detect, evaluate, and respond to user input by triggering each of the recited display actions as appropriate, depending on whether the input “meets” or “does not meet” specified criteria. Based on the plain claim language, a POSITA would have understood that logic operation to include the following steps:

- (i) “detecting a second input by a second contact that includes movement across the display in a direction;”
- (ii) “in response to detecting the second input,” determining whether “the second input meets” or “does not meet” the recited criteria; and
- (iii) “in accordance with” that yes/no determination, triggering performance of each corresponding display action where appropriate.

41. In the context of the '578 Patent's claims, a POSITA would have understood that in order to trigger a device to perform alternative display actions depending on whether an input either meets or does not meet specified criteria, the same programming logic must evaluate both instances of the user input (i.e., the input that “meets” and the input that “does not meet” the criteria). However, as I will further detail in Section X.A below, the Petition contradicts these basic principles—and, as a result, deviates from the plain and ordinary meaning of the claim language—by mapping Limitations [1j]-[1k] to two separate logic operations

that detect, evaluate, and respond to user input in different operational contexts. *See* Pet., 17-24. In particular, the Petition maps programming logic from Shiplacoff’s “**full-screen mode**” to Limitation [1j], alleging that the recited function is triggered when the device is operating in “full-screen mode” and detects user input that “meets” the specified criteria. Pet., 17-23. The Petition maps separate programming logic from Shiplacoff’s “**card mode**” to Limitation [1k], alleging that the recited function is triggered when the device is operating in “card mode” and detects user input that “does not meet” the specified criteria. *See id.*, 23-24. But that reading of the claims is inconsistent with the plain language of the claims, as well as the nature of programming logic.

42. Executable instructions to cause a device to perform a specific action require logic operations that detect when a user’s contact with the touchscreen matches the attributes for the gesture or input defined to trigger performance of the action. This type of logic operation is commonly referred to as an “event listener,” which is a piece of code designed to initiate a specific action in response to a defined event—such as a mouse click, key press, or a specific touchscreen interaction with specific attributes. An event listener can also respond to a sequence of events or to multiple events occurring simultaneously. Without such logic operations, the device would have no way of determining when to initiate the corresponding action, nor what corresponding action should be triggered. Indeed, from a software engineering

standpoint, it would not be possible to implement a function that performs an action in response to a defined input unless that function is designed to be executed because that defined input has been detected.

43. Here, the plain claim language dictates that triggering the performance of the display actions recited in Limitations [1j]-[1k] requires “detecting a second input by a second contact that includes movement across the display in a direction” while the device is “displaying the user interface of the application.” ’578 Patent, Limitation [1i]. The plain claim language further dictates that, to determine the display action to be performed, the device must evaluate “*the* second input” to determine whether it either “meets” or “does not meet” “one or more criteria that is met when the second input is detected to include a movement parameter that is above a movement threshold.” *Id.*, Limitations [1j]-[1k]. A POSITA would have interpreted this flow as a control structure in which execution of an action is governed by the context and evaluated attributes of user input.

44. Although the programming logic of Limitations [1i]-[1k] would not trigger the display actions recited in both Limitations [1j] and [1k] in response to the same instance of input—because an input cannot simultaneously “meet” and “not meet” the recited criteria—Limitations [1j] and [1k] nonetheless require programming logic capable of triggering each of the recited display actions. Thus, Limitations [1j]-[1k] are not satisfied by programming logic capable of triggering

performance of only one of the two recited display actions. Nor are they satisfied by separate logic operations that detect and evaluate user input in different operational contexts to trigger each of the recited display actions.

45. Rather, the plain language of Limitations [1i]-[1k] requires a single logic operation (or series of logic operations) executed while the device is “displaying the user interface of the application” and capable of: (i) “detecting a second input by a second contact that includes movement across the display in a direction,” (ii) determining whether “the second input” “meets” or “does not meet” the recited criteria, and (iii) triggering performance of each of the corresponding display actions as appropriate. ’578 Patent, Limitations [1i]-[1k].

46. As I explained above in Section VI.B, Claim 27 recites executable “instructions” that cause the device to perform logic operations that are substantially similar to those of Claim 1. Claim 27 differs from Claim 1 in that the logic operation recited in Limitations [27d]-[27f] specifies two different “criteria” against which the “second input” is evaluated. ’578 Patent, Limitations [27d]-[27f]. As with Claim 1, a POSITA would have understood Limitations [27d]-[27f] to require a single logic operation (or series of logic operations) executed in the operational context required by the claim—i.e., while the device is “displaying the user interface of the application”—that enables the device to: (i) “detect[] a second input by a second contact that includes movement across the display in a direction,” (ii) determine

whether “the second input” “meets” or “does not meet” the two recited criteria, and (iii) trigger performance of each of the recited display actions as appropriate. *Id.*, Limitations [27d]-[27f].

47. This interpretation of the plain language of Claims 1 and 27 is consistent with the descriptions in the ’758 Patent and ’918 Provisional of “decision-tree” logic for detecting pattern-based gestures and altering the output from and/or processing by the device in response thereto, which I discussed above in Section VI.B.

2. Claim 15

48. Limitations [15e]-[15g] of Claim 15 are substantially similar to Limitations [1i]-[1k] of Claim 1—indeed, the Petition makes no distinction between these limitations and instead merely incorporates the Petition’s contentions with respect to Limitations [1i]-[1k] for Limitations [15e]-[15g]. Pet., 33-34, 47. Limitations [15e]-[15g] should thus be interpreted consistently with Limitations [1i]-[1k].

49. Nonetheless, in the Institution Decision, the Board questioned whether the method steps of Limitations [15f]-[15g] “are written as mutually exclusive alternatives,” and specifically whether only “one of these two limitations needs to be satisfied in order to meet the claim.” ID, 7-8 (citations omitted).

50. The plain and unambiguous language requires that, to meet the method

of Claim 15, *each* of the display actions recited in Limitations [15f] and [15g] must be performed where their recited preconditions apply—i.e., where the touch input either meets or does not meet specified criteria. A POSITA would have understood this to be true even though Limitations [15f] and [15g] cannot both occur in response to the same instance of input. Each condition—i.e., meeting or not meeting the specified criteria—leads to a different display action, and understanding the claim fully requires evaluating all possible conditions and their corresponding display actions.

51. Further, Claim 15 recites that the method must be performed “at a mobile device” and therefore explicitly requires “a mobile device” that is configured to perform each display action when the corresponding preconditions are met. *See* ’578 Patent, Limitation [15a]. Thus, like with Claims 1 and 27, a POSITA would have understood Claim 15 to require a single logic operation (or series of logic operations) executed in the operational context required by the claim—i.e., while the device is “displaying the user interface of the application”—that enables the device to: (i) “detect[] second a second input by a second contact that includes movement across the display in a direction,” (ii) determine whether “the second input” “meets” or “does not meet” the two recited criteria, and (iii) trigger performance of each of the corresponding display actions as appropriate. *Id.*, Limitations [15f]-[15g].

52. This interpretation of the plain language of Claim 15 is consistent with

the descriptions in the '578 Patent and '918 Provisional of “decision-tree” logic for detecting pattern-based gestures and altering the output from and/or processing by the device in response thereto, which I discussed above in Section VI.B.

B. “The Second Input”

53. As I will detail further in Section X.C.1 below, the Petition also attempts to deviate from the plain language of the claims by mapping touch events detected and evaluated in two different operational contexts to the claimed “second input.” *See* Pet., 15-24. In particular, the Petition maps “the second input” of Limitation [1j] to the portion of Shiplacoff describing a “swipe up” gesture that is detected in “*full-screen mode*.” Pet., 17-23. The Petition then maps “the second input” of Limitation [1k] to the portion of Shiplacoff describing a “tap or flick” that is detected while in a “*card mode*.” *See id.*, 23-24.

54. But the plain language of the claims require that “the second input” of Limitations [1j] and [1k] be detected *in the same operational context*. In particular, Limitation [1i] recites: “while displaying the user interface of the application, detecting a second input by a second contact that includes movement across the display in a direction.” And Limitations [1j]-[1k] collectively recite executing a logic operation “in response to detecting the second input” to (a) determine whether “the second input” either “meets” or “does not meet” specified criteria, and (b) “in accordance with” that yes/no determination, trigger each of the recited display

actions where appropriate, as I explained above in Section VIII.A. Limitations [15f]-[15g] and [27d]-[27f] include similar language. The word “the” dictates that Limitations [1i]-[1k], [15f]-[15g], and [27d]-[27f] require a single, common contact event as the starting point of the claimed logic operations. And the plain language of the claims define that common starting point as “a second input by a second contact that includes movement across the display in a direction,” which is detected while the device is “displaying the user interface of the application.” ’578 Patent, Limitations [1i], [15e], [27d].

55. As I explained above in Sections VI.B and VIII.A, each use of the phrase “*the* second input” in Limitations [1j]-[1k], [15f]-[15g], and [27e]-[27f] similarly refers to the antecedent phrase “*a* second input” in Limitations [1i], [15e], and [27d]. *See, e.g.*, Terveen Deposition, 56:22-57:21 (“Each of [1j] and [1k] refer back -- the mention of ‘the second input’ in each of them is referring back to a second input that is mentioned in Limitation [1i].”), 58:22-59:13 (answering similarly for Claims 15 and 27). Thus, as a POSITA would have understood, the claim cannot be met by a user input that meets only “*the* second input” in Limitations [1j]-[1k], [15f]-[15g], and [27e]-[27f]; rather, this same user input must also satisfy the requirements for “*a* second input” in Limitations [1i], [15e], [27d]. *See* Terveen Deposition, 57:14-59:13 (agreeing that “*the* second input” “must be detected while the user interface of the application is displayed and must include movement across the

display in a direction”).

56. As I explained above in Section VI.C, Dependent Claims 2-3, 16-17, and 28-29 also refer to, and place further constraints on, “*the* second input.” In particular, Dependent Claims 2, 16, and 28 recite that “*the* second input” is by a contact “that includes movement that starts at an edge of the display.” Terveen Deposition, 59:17-61:7 (agreeing that claim “dependency requires all of the limitations of [the independent claim] as well as the one added by [the dependent claim]” and that Dependent Claims 2, 16, and 28 refer to “the same second input that is first introduced in” Limitations [1i], [15e], and [27d]). Dependent Claims 3, 17, and 29 recite that “*the* second input” is by a contact “that includes movement across the display in... a vertical direction.” *Id.* (agreeing similarly with respect to these claims). Again, the word “the” indicates that, in each instance of operation, these claims impose additional requirements on “*the* second input” in the independent claims. *See id.*

IX. OVERVIEW OF THE PRIOR ART

A. Shiplacoff

57. Shiplacoff discloses a device in which multiple “[display] modes are available for interacting with [the] device.” Shiplacoff, ¶[0068]. One such “display mode” is “full-screen mode,” in which (as shown in FIG. 2, which I have reproduced below) an “activity” such as a “calendar application occupies substantially the entire

display screen.” *Id.*, ¶[0075].

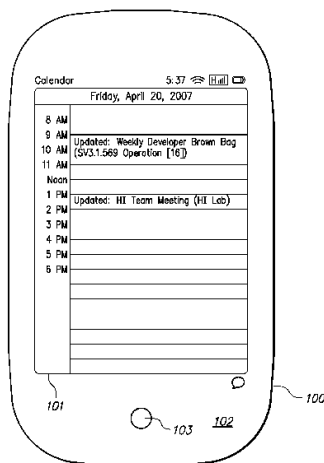


FIG. 2

Id., FIG. 2. “[F]ull-screen mode is optimized for maximum use of screen space for interacting with a single activity.” *Id.*, ¶[0074]; Terveen Deposition, 23:21-24:8, 26:1-7 (acknowledging the same). Shiplacoff discloses that “full-screen” mode allows the user to “interact[] with the active activity (such as the calendar application shown in FIG. 2)” by “manipulating on-screen elements” within the application. Shiplacoff, ¶¶[0074]-[0076].

58. Among the display modes disclosed by Shiplacoff is “card mode.” *Id.*, ¶¶[0068]-[0074]. Unlike “full-screen mode,” “card mode is optimized for easily switching focus from one activity to another and for easily initiating and terminating activities.” *Id.*, ¶[0074]; Terveen Deposition, 24:4-8 (agreeing with the same); *see also* Terveen Deposition, 23:3-8 (agreeing that “card mode is optimized for something different than what full-screen mode is optimized for”). Shiplacoff

discloses that, to achieve this, applications or other activities are represented by “cards” “that can be moved and manipulated by the user” on the screen. Shiplacoff, ¶[0060]. Multiple of such “cards” are displayed when in “card mode,” “each card representing an application or other activity.” *Id.*, ¶¶[0068]-[0072]; Terveen Deposition, 31:1-6 (“That is something that card mode, shuffle mode, and grid mode all have in common, that multiple cards are shown at once.”). Shiplacoff’s FIG. 3 illustrates an example in which three cards 301A-301C representing three different applications are displayed by the device while in “card mode.” Shiplacoff, ¶¶[0078]-[0079].

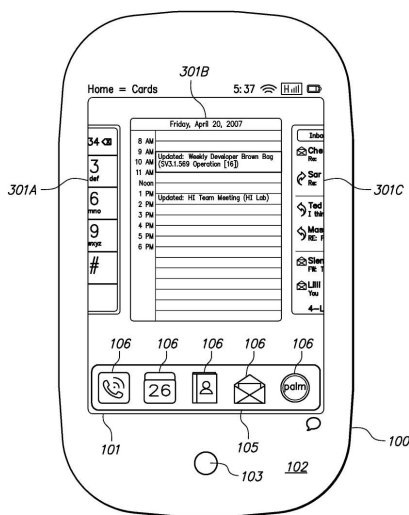


FIG. 3

Id., FIG. 3.

59. Shiplacoff discloses that, when in “card mode,” a user can rearrange, dismiss, and switch between cards using various on-screen commands. *Id.*,

¶¶[0084]-[0106]. For instance, Shiplacoff discloses that a user can perform a “flicking action” on a card upward off the screen to dismiss the card and its associated application. *Id.*, ¶¶[0100]-[0103]. The user can also perform a different input on the card—such as a “touch or double-tap”—to reenter a “full-screen mode” that displays and reenables interaction with the card’s application. *Id.*, ¶[0091]; Terveen Deposition, 24:16-25:2 (“[I]n card mode you cannot have a focus on a single activity to interact with, so you would have to change modes if you want to do that.”); *see also, e.g.*, Shiplacoff, ¶[0128] (explaining that “a ‘swipe down’ gesture beginning on screen 101 and extending onto gesture area 102” is another input that can switch “from card mode to full-screen mode”).

60. Another display mode described by Shiplacoff is “shuffle mode.” Shiplacoff, ¶¶[0068]-[0074]. Although Shiplacoff describes “shuffle mode” as a separate display mode from “card mode,” Shiplacoff explains that “the user can interact with cards 301 in shuffle mode in substantially the same manner as in card mode.” *Id.*, ¶[0112]; Terveen Deposition, 29:2-22, 31:1-6 (confirming similarities in interaction between “card mode” and “shuffle mode”).

61. Shiplacoff discloses that various user inputs are programmed to trigger a switch between the disclosed display modes or a switch between certain display modes and the device’s “launcher screen.” *See, e.g.*, Shiplacoff, ¶¶[0099], [0124]-[0132]. For instance, Shiplacoff discloses that a user can switch from “card mode”

to the device's "launcher screen" "by performing a gesture or other command." *Id.*, ¶[0099]. Shiplacoff also discloses that the user can initiate a switch from "full-screen mode" to "card mode" by using "a 'swipe up' gesture" that begins in an *off-screen* "gesture area" and extends onto the screen. *Id.*, ¶[0127].

62. Shiplacoff discloses—through its incorporation by reference of Duarte-992 (*see* Shiplacoff, ¶[0066])—that the device is able to distinguish the described "swipe up" gesture from other inputs based on whether the gesture or input includes contact with the off-screen "gesture area 102" (green, below) and/or a cross-over between "gesture area 102" and "screen 101" (red, below). Duarte-992, ¶¶[0059], [0077], Abst.

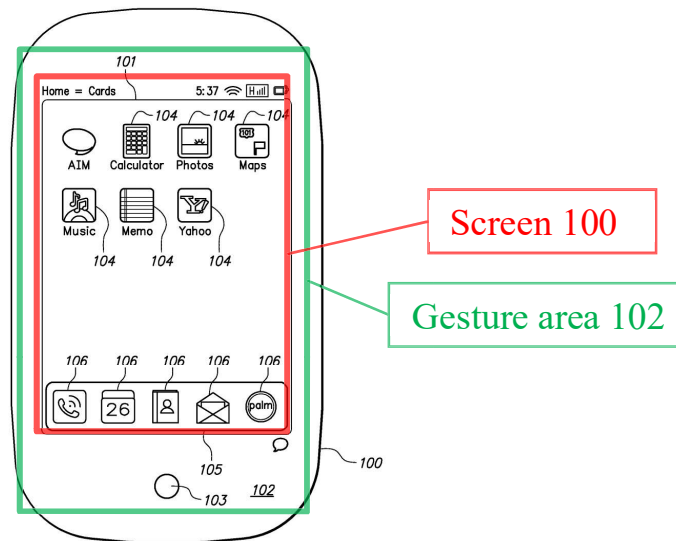
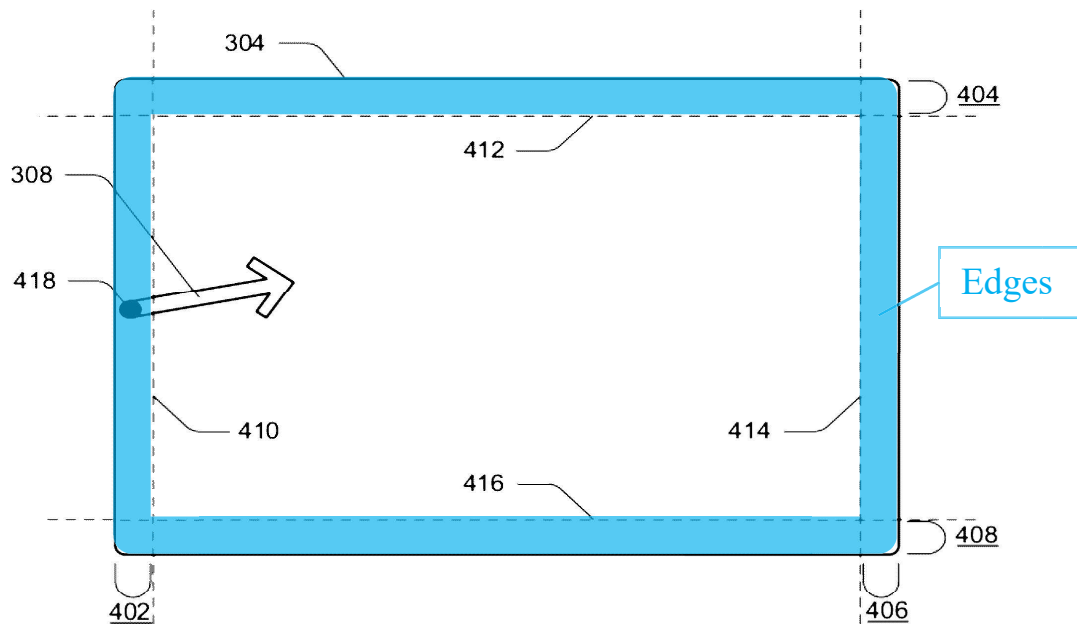


FIG. 1

Shiplacoff, FIG. 1 (annotated); *see also* Duarte-992, ¶¶[0060], [0062].

B. Nan

63. Nan discloses the use of an “edge gesture” that “has a start point at an edge of [a] user interface and a later point not within the edge.” Nan, Abst., 7:17-22. Nan explains that the “edge” for an “edge gesture” is a “small” region of pixels that extends inward up to an “edge limit,” as shown in my annotated version of Nan’s FIG. 4 provided below. *Id.*, 4:35-62.



Id., FIG. 4 (annotated). Nan’s FIG. 4 illustrates four “edges” 402, 404, 406, and 480 (highlighted in blue), each of which lie between the “display limit” of an “immersive interface 304” and an inner “edge limit” (dashed lines 410, 412, 414, and 416). *Id.*, 4:52-62. Using these “edges,” an “edge gesture” is distinguished from other user inputs based on whether the “start point” of a touch event “is at an edge.” *Id.*, 4:35-5:2, 4:65-5:2.

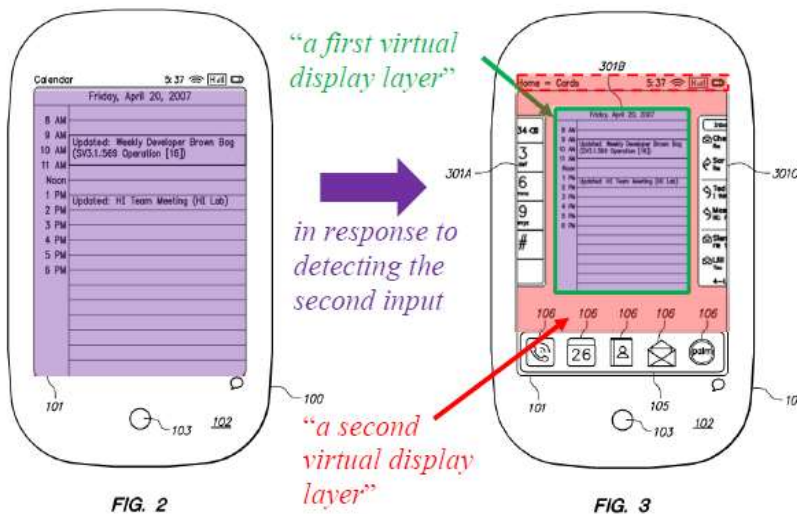
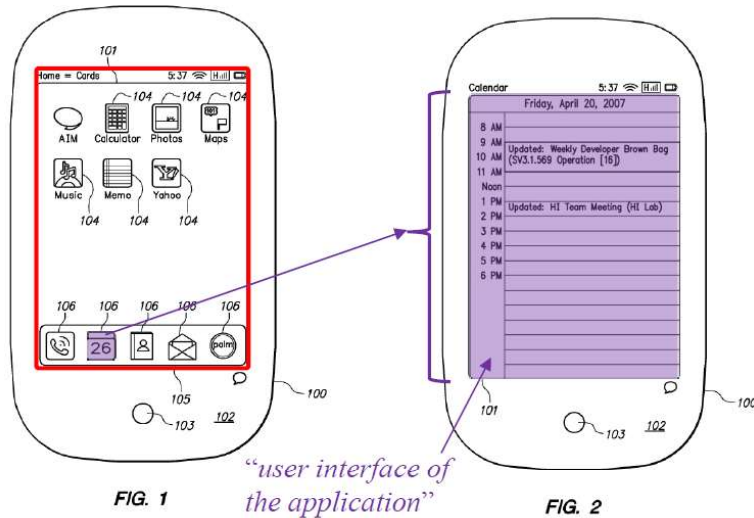
X. THE PETITION’S GROUND 1 FAILS TO ESTABLISH THAT THE PRIOR ART DISCLOSES THE CHALLENGED CLAIMS

A. Limitations [1i]-[1k], [15e]-[15g], [27d]-[27f]: Shiplacoff Fails to Disclose or Suggest Programming Logic Capable of Triggering the Recited Display Actions in the Required Operational Context

64. As I explained above throughout Sections VI.B and VIII.A-VIII.B, Limitations [1i]-[1k], [15e]-[15g], and [27d]-[27f] require programming logic capable of evaluating “a second input” detected in a particular operational context—while “displaying the user interface of the application”—to determine whether “the second input” “meets” or “does not meet” specified criteria and triggering each of the recited display actions as appropriate “in accordance with” that determination. The word “the” dictates that any input mapped to Limitations [1j]-[1k], [15f]-[15g], and [27e]-[27f] must be detected while the device is “displaying the user interface of the application,” as I discussed above in Section VIII.B.

65. The Petition’s Ground 1 maps Shiplacoff’s “swipe up” gesture to “the second input” of Limitations [1j], [15f], and [27e] and maps Shiplacoff’s “tap or flick” input to “the second input” of Limitations [1k], [15g], and [27f]. Pet., 15-24, 36, 38-39. The Petition’s mapping fails, however, because Shiplacoff discloses that its “swipe up” gesture and the “tap or flick” inputs are utilized—and therefore detected—in two different operational contexts, only one of which the Petition maps to the operational context required by the claims—“while displaying the user interface of the application.”

66. Shiplacoff discloses that its “swipe up” gesture is utilized when the device is operating in “*full-screen mode*”—such as the full-screen display of a “calendar user interface” as shown in Shiplacoff’s FIG. 2—and causes the device to change the display from “full-screen mode” to “card mode.” Shiplacoff, ¶[0127]; Terveen Deposition, 35:18-16 (agreeing that Shiplacoff’s ¶[0127] includes an “example[] listed three times” of switching “from full-screen mode to card mode” in response to a “swipe up” gesture). The Petition maps Shiplacoff’s “full-screen mode” to the claimed “while displaying the user interface of the application,” which Limitations [1i], [15e], and [27d] dictate is the operational context in which “the second input” must be detected. Pet., 15-23; Terveen Deposition, 61:22-63:2 (“[T]he thing that I’m saying maps to the user interface of the application, the application I’m referring to is[,] Shiplacoff discloses that by an application displayed in full-screen mode.”). The Petition maps “card mode” to “display[ing] at least a portion of the user interface of the application in a first virtual display layer” when “the second input,” which is the display action of Limitations [1j], [15f], and [27e]. Pet., 17-23; Terveen Deposition, 63:3-64:8 (explaining that, for the claimed “displaying at least a portion of the user interface of the application in a first virtual layer, I [(Dr. Terveen)] was talking about what happens when you now have gone from full-screen mode of an application to card mode”). This mapping is illustrated in the Petition’s annotations to Shiplacoff’s figures, which I have reproduced below:



Pet., 14-15 (annotating, for Limitation [1h], Shiplacoff’s FIGS. 1-2), 16 (asserting that because “Shiplacoff’s device detects a gesture, such as a swipe up gesture, while displaying an application on the touch screen display, Shiplacoff teaches [Limitation [1i]]”), 17-18 (annotating Shiplacoff’s FIGS. 2-3 and arguing that, “[a]s explained in [1i], a user can switch from the full-screen mode to the card mode... by using a swipe up gesture”), 36-37.

67. As for Shiplacoff’s “tap or flick” input that the Petition maps to

Limitations [1k], [15g], and [27f] (*see* Pet., 23, 39), Shiplacoff discloses that when the device is operating in “**card mode**,” “a card can be dragged or flicked off screen, or a control can be selected to close a display card and, in some cases, return to the launch screen.” Shiplacoff, ¶[0100]. But, as I explained above throughout Sections VI.B and VIII.A-VIII.B, the claims dictate that “the second input” of Limitations [1j]-[1k], [15f]-[15g], and [27e]-[27f] must be detected “while displaying the user interface of the application,” which the Petition maps to Shiplacoff’s “full-screen mode.” *See* Pet., 15-23; Terveen Deposition, 61:22-63:2. And as the Petition acknowledges, Shiplacoff’s “card mode” is different from “full-screen mode.” Pet., 6-7; Terveen Deposition, 20:7-20 (agreeing that “card mode” is “different from the full-screen mode in that... there’s three cards shown”).

68. The Petition attempts to bridge this gap by focusing on the following two sentences from Shiplacoff:

In one embodiment, the user can launch additional activities by returning to the launch screen as shown in FIG. 1. In one embodiment, this is performed by performing a gesture or other command.

Shiplacoff, ¶[0099]; Pet. 23-24. The Petition argues these two sentences suggest that Shiplacoff’s system is able to distinguish “a swipe up gesture” from “a tap or flick,” and, in the instance of detecting the latter in “full-screen mode,” would close the displayed activity (such as the “calendar application” illustrated in FIG. 2) to switch from the “full-screen mode” to the “launch screen in FIG. 1.” Pet., 23-24.

69. But I find nothing in Shiplacoff’s ¶[0099] or the other paragraphs cited in the Petition (*see* Pet., 23-24 (citing Shiplacoff, ¶¶[0099]-[0102], [0167])) suggesting that Shiplacoff’s “tap or flick” input is utilized in Shiplacoff’s “**full-screen mode**.” To the contrary, the cited paragraphs of Shiplacoff address only the device’s responses to user input while in “**card mode**” and/or “**shuffle mode**.”

70. In particular, Shiplacoff’s paragraphs [0099]-[0102] are included in a section of the specification entitled “Card Mode.”² Shiplacoff, ¶¶[0077]-[0106]. In all preceding paragraphs of Shiplacoff’s “Card Mode” section, Shiplacoff describes the device’s operation with respect to entering and exiting “card mode.” *See generally id.*, ¶¶[0077]-[0098]. Paragraphs [0097]-[0098]—i.e., the portion of Shiplacoff’s disclosure immediately preceding the portion of ¶[0099] on which the Petition focuses—describe how Shiplacoff’s device enters, operates in, and exits “card mode” as illustrated across Shiplacoff’s FIGS. 6A-6F. *Id.*, ¶¶[0097]-[0098]. The remainder of ¶[0099] and all paragraphs that follow within Shiplacoff’s “Card Mode” section likewise describe the device’s operation when in “card mode,” as

² Although Shiplacoff includes a separate section entitled “Full Screen Mode,” that section makes no reference to Shiplacoff’s “tap or flick” input on which the Petition relies. Shiplacoff, ¶¶[0075]-[0076].

illustrated in each of Shiplacoff's FIGS. 7, 8A-8B, and 9A-9D. *See generally id.*, ¶¶[0099]-[0106]. In particular, these disclosures focus on how “newly launched activities generate new cards that are added to the end of the sequence of cards shown in FIG. 7” (*id.*, ¶[0099]; Terveen Deposition, 27:16-28:9 (agreeing that FIG. 7 is “a conceptual depiction of what’s happening in card mode”)), how “a card 301 can be closed by,” e.g., “clicking on a control within the card[,]... dragging the card upward off screen 101, or performing a flicking action in an upward direction” as shown in FIGS. 8A-8B (Shiplacoff, ¶¶[0100]-[0103]; Terveen Deposition, 28:10-29:1 (agreeing that FIGS. 8A-8B depict “an embodiment for dismissing or closing a card in card mode”), and how cards can be “rearrange[d]... while in card mode[] by tapping and holding... and then dragging a card 301 from one position to another” as shown over FIGS. 9A-9D (Shiplacoff, ¶¶[0104]-[0106]). Nowhere in Shiplacoff's ¶¶[0099]-[0102] is “full-screen mode” discussed or mentioned. Given that context, a POSITA would have viewed ¶¶[0099]-[0102] as limited to *the specific operational context of “card mode.”*

71. Shiplacoff's ¶[0167]—which is located in a section of Shiplacoff entitled “Card Groups” (Shiplacoff, ¶¶[0155]-[0184])—is similarly limited. The title of the section in which ¶[0167] is included alone indicates that ¶[0167] relates to Shiplacoff's “card mode.” Indeed, ¶[0167] begins by explaining that, “*/i/n shuffle mode and/or in card mode*, a user can remove a card 301 from a group by

dragging the card 301 to an area of the screen not occupied by a card group.” *Id.*, ¶[0167]. The paragraph goes on to explain Shiplacoff’s FIGS. 17A-17D, which further elaborate on this “shuffle mode” and/or “card mode” operation. *Id.*, ¶[0167]; Terveen Deposition, 29:2-22 (agreeing with the same). And like ¶¶[0099]-[0102], Shiplacoff’s ¶[0167] does not discuss or mention “full-screen mode.”

72. The Petition cherry-picks statements from Shiplacoff without giving proper consideration to these statements’ context. A POSITA considering Shiplacoff as a whole would have understood the cited paragraphs of Shiplacoff to disclose only detecting—and taking action in response to—Shiplacoff’s “tap or flick” input while the device is in “*card mode*” and/or “*shuffle mode*.” Shiplacoff, ¶¶[0099]-[0102], [0167]; *see also* Terveen Deposition, 47:14-49:10 (acknowledging that “gestures are assigned a meaning depending on the context in which they’re used” and that “a POSITA implementing Shiplacoff would understand that not all modes are going to use a given gesture in the same way”).

73. Because the Petition fails to identify any disclosure or suggestion in Shiplacoff that the “tap or flick” described in ¶¶[0099]-[0102] and [0167] is detected while in “full-screen mode”—which is the only display mode the Petition maps to

“while displaying the user interface”—the Petition’s Ground 1 mapping fails.³

B. Limitations [1j]-[1k], [15f]-[15g], [27e]-[27f]: Shiplacoff Does Not Disclose or Suggest Using a “Movement Threshold” To Differentiate Between Inputs

74. As I explained throughout Sections VI.B and VIII.A-VIII.B above, Limitations [1j]-[1k], [15f]-[15g], and [27e]-[27f] require programming logic capable of evaluating “the second input” to trigger each of the recited display actions as appropriate depending on whether “the second input” “meets” or “does not meet” “one or more criteria that is met when the second input is detected to include a movement parameter that is above a movement threshold.” The Petition argues it would have been obvious to a POSITA that “Shiplacoff’s device can distinguish different gestures.” Pet., 23-24. But the only support the Petition offers is that Shiplacoff apparently discloses a device that “detects user input and differentiates between contacts and gestures such as a single tap or swipe up gesture, so that it can

³ In addition, as I will explain below in Section X.C.3, the Petition’s Ground 1 mapping also fails because the Petition does not identify any disclosure or suggestion that a “tap or flick” input would cause the device to switch the display from “full-screen mode” to the “launch screen,” which the Petition maps to the recited display action of Limitations [1k], [15g], and [27f].

determine the appropriate corresponding operation to execute.” *Id.*, 21-22.

75. Even if that were an accurate characterization of Shiplacoff—and I do not agree that it is—distinguishing between an input that does *not* have movement (such as a tap) and one that does have movement (such as a swipe) is insufficient to disclose Limitations [1j]-[1k], [15f]-[15g], and [27e]-[27f]. Those limitations recite that “the second input” includes “movement across the display” and further recite determining whether that input includes “a movement parameter above a movement threshold, as I explained above in Sections VI.B and VIII.A-VIII.B Thus, the plain language of the claims requires programming logic to distinguish between two inputs, both of which include movement across the display, based on whether the movement parameter of the input is above or below a movement threshold.

76. Shiplacoff does not disclose or suggest distinguishing between two inputs on that basis. Instead, Shiplacoff discloses distinguishing between inputs that involve movement across the display (such as its “swipe up” gesture or “flick” input) based on whether the input does or does not include contact with an off-screen “gesture area.” Shiplacoff, ¶[0066].

77. As shown in my annotated version of Shiplacoff’s FIG. 1 provided below, the device described in Shiplacoff includes a “touch-sensitive gesture area 102” (green) that “extend[s] beyond the edges of touch-sensitive screen 101” (red):

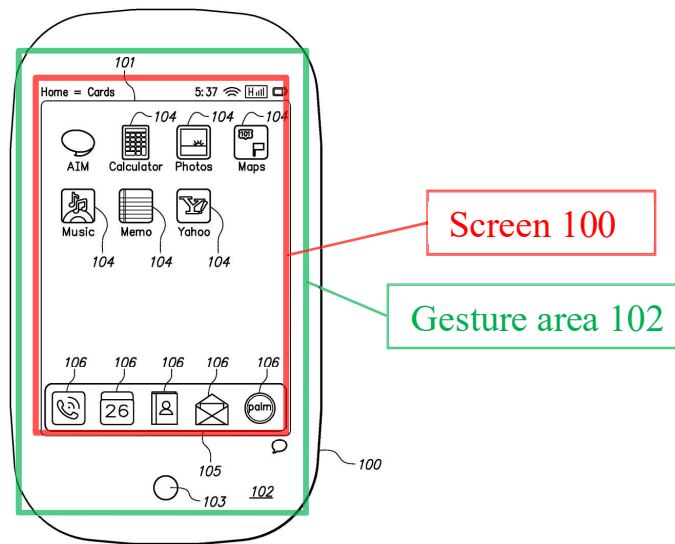


FIG. 1

Id., FIG. 1 (annotated), ¶¶[0064],[0066]; *see also* Duarte-992, ¶¶[0060], [0062].

78. When describing how the device uses these “touch-sensitive” areas to distinguish between different detected gestures and to activate the appropriate corresponding function, Shiplacoff refers to Duarte-992, which Shiplacoff incorporates by reference. Shiplacoff, ¶¶[0066], [0064]. Duarte-992 discloses that the device “allows the user to enter touch commands such as gestures in gesture area 102 and/or touch-sensitive screen 101, as well as to enter touch commands that ***cross over*** from gesture area 102 to touch-sensitive screen 101, and vice versa.” Duarte-992, ¶[0059]; *see also* Terveen Deposition, 73:19-74:5 (acknowledging consistent item numbering between Shiplacoff and Duarte-992). The device then evaluates whether the command “includes one ***or both of these areas***, and can perform whatever action is appropriate to such input.” Duarte-992, ¶[0013].

79. For example, Shiplacoff discloses that its “swipe up” gesture “begin[s] in gesture area 102 and extend[s] onto screen 101.” and a “swipe down” gesture “begin[s] on screen 101 and extend[s] onto gesture area 102.” Shiplacoff, ¶¶[0127]; *see also* Duarte-992, ¶[0077]. A POSITA would have understood from this disclosure that Shiplacoff’s device differentiates its “swipe up” gesture from its “flick” input⁴ based on the fact that the “swipe up” gesture begins in the gesture area and includes “a cross over” from gesture area 102 to touch-sensitive screen 101.

80. Neither Shiplacoff nor Duarte-992 discloses or suggests distinguishing between user inputs based on a movement threshold. Instead, Duarte-992 discloses “implementing a vocabulary of touch commands including gestures that are performed within gesture area 102, within screen 101, *or on some combination of the two*” in order to “expand[] the available space and the vocabulary of gestures over prior art systems.” Duarte-992, ¶[0072]. In other words, it is the “*gesture area/]*” [that] allows a user to activate commands using a gesture vocabulary.” *See* Duarte-992, Abst. Accordingly, a POSITA would have understood Shiplacoff to disclose differentiating user input based on whether the input includes contact with

⁴ As I will explain below in Sections X.C.1-X.C.2, Shiplacoff defines its “flick” input as being an input that begins on-screen.

Shiplacoff's gesture area, rather than based on whether the input includes a "movement parameter that is above a movement threshold."

81. Because Shiplacoff does not disclose or suggest using a "movement threshold" to differentiate between inputs, Shiplacoff does not disclose Limitations [1j]-[1k], [15f]-[15g], and [27e]-[27f].

C. Limitations [1k], [15g], [27f]: The Petition's Mapping of Shiplacoff's "Tap or Flick" Input Fails for Several Reasons

82. The Petition's mapping for Limitations [1k], [15g], and [27f] is premised on Shiplacoff's disclosures regarding the use of its "tap or flick" input, which the Petition argues causes the device to "return[] to the launch screen" when in "full-screen mode." Pet., 23-24. As I will explain below, however, that mapping fails because: (1) Shiplacoff does not disclose or suggest that its "tap or flick" input triggers a switch from "full-screen mode" to "the launch screen," (2) a POSITA would not have been motivated to modify Shiplacoff to do so, and (3) a "tap" input does not "include[] movement across the display" and therefore cannot be "the second input."

1. Shiplacoff Does Not Disclose or Suggest That Its "Tap or Flick" Input Triggers the Recited Display Action

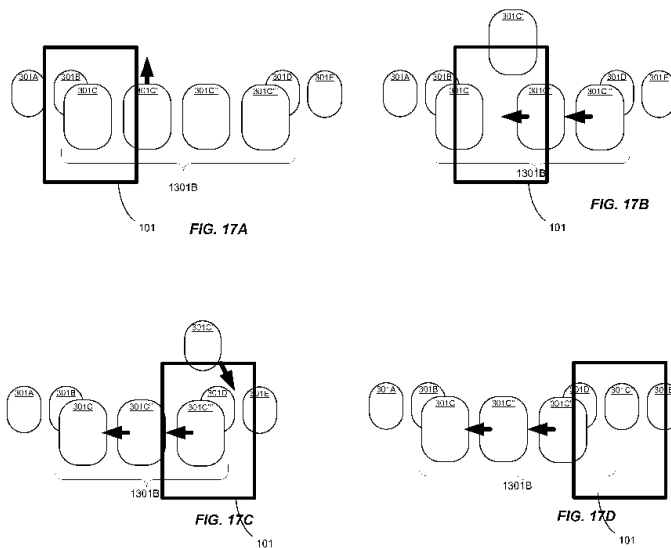
83. As I explained above throughout Sections VI.B and VIII.A-VIII.B, the display action recited in Limitations [1k], [15g], and [27f] is "replacing the user interface of the application with the home screen including the icon associated with

the application.” The Petition maps the claimed “user interface of the application” to Shiplacoff’s “full-screen mode” and the claimed “home screen” to Shiplacoff’s “launcher screen.” Pet., 16-23; Terveen Deposition, 61:22-63:2, 64:9-65:6. But the Petition fails to identify any disclosure in Shiplacoff that its “tap or flick” input triggers a switch from “full-screen mode” to the “launcher screen.”

84. As I explained above in Section X.A, the portions of Shiplacoff that the Petition cites with respect to Shiplacoff’s “tap or flick” input (i.e., paragraphs [0099]-[0102] and [0167]) are limited to the device’s response to such input while in the specific operational context of “*card mode*” and/or “*shuffle mode*.” I additionally explained in Section X.A that the cited portions of Shiplacoff relate to the use of its “tap or flick” input in these modes to close cards, *not* to switch from “full-screen mode” to another or to return to the “launcher screen.” And because Shiplacoff discloses that display modes are optimized for different purposes, a POSITA would have understood that an input used to trigger an action in one display mode—such as Shiplacoff’s “tap or flick” input in “card mode”—would not be programmed to trigger the same action in a different display mode. See Terveen Deposition, 39:6-21 (agreeing that, “when you’re in full-screen mode, certainly that is optimized for interaction with a particular activity, like your calendar application or your email or whatever it may be,” “[a]nd when you’re using gestures to interact in general with your device, some of those gestures will be focused on that activity”).

85. For example, Shiplacoff discloses that “card mode is optimized for easily switching focus from one activity to another and for easily initiating and terminating activities.” Shiplacoff, ¶[0074]; Terveen Deposition, 24:4-8. To achieve this, “card mode” displays multiple “cards” on-screen that each represent an “activity” and enables the user to move cards (either rearranging them on the screen or moving them off the screen) “so as to change focus from one card to another.” Shiplacoff, ¶[0015]; Terveen Deposition, 22:8-17 (agreeing that, in “card mode,” “you have some way of navigating through the cards and changing focus from one another,” including “scrolling left and right”), 31:1-6 (agreeing that, in “card mode,” “multiple cards are shown at once”). Shiplacoff discloses that “shuffle mode” is similarly “optimized for reordering and manipulating cards representing activities.” Shiplacoff, ¶¶[0074], [0016]; Terveen Deposition, 30:12-31:6 (acknowledging this disclosure).

86. Within this context, a POSITA would have understood Shiplacoff to disclose that its “tap or flick” input is an on-screen input available in “card mode” that “cause[s] cards to move on the screen, so as to change focus from one card to another.” Shiplacoff, ¶¶[0015]-[0016], [0112]. This is illustrated in Shiplacoff’s FIGs. 17A-17D, which I have reproduced below:



Id., 17A-17D, ¶[0167]; *see also id.*, ¶[0112].

87. In contrast, Shiplacoff discloses that “in a full-screen mode, one activity occupies substantially an entire display screen” and “[t]he user interacts with the active activity (such as the calendar application shown in FIG. 2) by conventional user interface methods, including manipulating on-screen elements, entering text, and the like.” *Id.*, ¶¶[0014], [0076]; *see also* Terveen Deposition, 23:3-8 (agreeing that “card mode is optimized for something different than what full-screen mode is optimized for”), 39:6-21 (acknowledging that, “when you’re in full-screen mode, certainly that is optimized for interaction with a particular activity,” and so “when you’re using gestures to interact in general with your device, some of those gestures will be focused on that activity”).

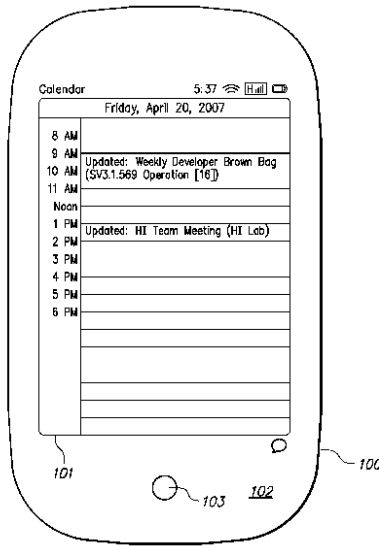


FIG. 2

Shiplacoff, FIG. 2. Given these disclosures, a POSITA would not have recognized Shiplacoff’s “tap or flick” input—which Shiplacoff describes as being an on-screen input for “manipulating *cards* representing activities” when in “card mode” (*id.*, ¶¶[0074], [0099]-[0101])—to be applicable to or available in “full-screen mode”—which is instead optimized for, e.g., “manipulating *on-screen elements*” *within* a single “active activity” (*id.*, ¶[0076]). *See also* Terveen Deposition, 26:1-17 (acknowledging “[f]ull-screen mode and card mode are different modes, different user interfaces. And in full-screen mode, you are *not* navigating through a set of different cards”), 24:16-25:2 (explaining that “in card mode you cannot have a focus on a single activity to interact with, so you would have to change modes if you want to do that”).

88. Although Shiplacoff discloses several mechanisms for “switching

among modes,” it does not disclose or suggest any gesture or touch-input that triggers switching from “full-screen mode” (which the Petition maps to the “user interface of the application” to the “launcher screen” (which the Petition maps to “the home screen”). In particular, in ¶¶[0124]-[0132], Shiplacoff describes various inputs that trigger switching between the four display modes (i.e., “full-screen,” “card,” “shuffle,” and “grid”), but describes switching to the “launcher screen” *only* from “card mode” and *only* in response to its “swipe up” or “swipe down” gestures. Shiplacoff, ¶¶[0124]-[0132]. Thus, the Petition fails to show that Shiplacoff discloses or suggests the use of its “tap or flick” input to trigger “replacing the user interface of the application with the home screen.”

89. The Petition alternatively argues that a POSITA would have found it obvious to use Shiplacoff’s “tap or flick” input to trigger “replacing the user interface of the application with the home screen.” Pet., 23-24. But that argument fails because, as I will explain below, the Petition fails to demonstrate that a POSITA would have been motivated to modify Shiplacoff to do so.

2. A POSITA Would Not Have Been Motivated To Modify Shiplacoff Such That Its “Tap or Flick” Input Triggers “Replacing the User Interface of the Application with the Home Screen”

90. The Petition does not identify any particular motivation to modify Shiplacoff such that Shiplacoff’s “tap or flick” input would trigger the recited display action of Limitations [1k], [15g], [27f]. Instead, the Petition argues merely

that it would have been “a matter of design choice” to use the “tap or flick” to “return the display... to the launch screen.” Pet., 24. Aside from Shiplacoff—which does *not* disclose or suggest using its “tap or flick” to switch from “full-screen mode” to the “launch screen” as I explained above in Section X.C.1—the Petition’s only support for this argument is the testimony of Dr. Loren Terveen, who merely repeats *verbatim* the Petition’s unsupported argument.

91. Neither the Petition nor Dr. Terveen explain why a POSITA would have been motivated to modify Shiplacoff to use its “tap or flick” input to trigger switching from “full-screen mode” to the “launch screen.” Rather, both the Petition and Dr. Terveen simply conclude this assertion to be true without any sufficient support.

92. In addition, the Petition fails to account for the other disclosures in Shiplacoff. Setting aside Shiplacoff’s “tap” input, which independently fails to satisfy the claims as I will explain below in Section X.C.3, Shiplacoff teaches away from using its “flick” input to trigger a switch from “full-screen mode” to the “launcher screen.”

93. For example, Shiplacoff discloses that its “full-screen mode” reserves substantially the entire user-interface area for, e.g., “manipulating *on-screen elements*” within a single “active activity.” Shiplacoff, ¶¶[0076], [0074]; Terveen Deposition, 39:6-21 (acknowledging that, in “full-screen mode,” “when you’re using

gestures to interact in general with your device, some of those gestures will be focused on that activity”). A POSITA would have recognized that, in order to program a device such that a particular input triggers a switch from “full-screen mode” to either the “launch screen” or one of the other described display modes, the input must be one that is not already programmed to trigger a different function or action related to the user’s manipulation of on-screen elements in the application. *See also* Terveen Deposition, 42:16-45:17 (agreeing that, “[i]n any given user interface or mode, a gesture should have a single meaning”), 49:11-51:16 (agreeing that “a POSITA implementing Shiplacoff is going to consider which gesture might be appropriate for a given mode” and would avoid, for instance, “us[ing] tapping for switching from the full-screen mode” because Shiplacoff’s “tap” “key[s] off the idea that there could be unoccupied areas of the screen or not occupied by any card” when in “card mode, shuffle mode, and grid mode”).

94. A POSITA would have understood that Shiplacoff’s “flick” input is already programmed to trigger a different action in “full-screen mode.” Shiplacoff explains that, in “full-screen mode,” “[t]he user interacts with the active activity (such as the calendar application shown in FIG. 2) by conventional user interface methods, including manipulating on-screen elements[.]” Shiplacoff, ¶[0076]. A number of such “conventional” techniques for “direct manipulation of on-screen objects” are disclosed by Shiplacoff through its incorporation by reference of

Duarte-992. Shiplacoff, ¶[0066]; Duarte-992, ¶¶[0013], [0006], [0080]. In particular, Duarte-992 discloses “a *drag-scroll* operation” in which a user performs “a *drag* gesture (also referred to as a *flick* gesture) across [the] screen in a direction that supports *scrolling* for the current state of the display.” Duarte-992, ¶[0089] (item numbering omitted). Thus, Shiplacoff—via Duarte-992—discloses that its “flick” input is an on-screen input that, when in “full-screen mode,” is already programmed to trigger “scrolling” of displayed objects. *See also* Terveen Deposition, 71:16-73:18 (admitting to not having seen or reviewed Duarte-992)

95. Even without these express disclosures, a POSITA would have understood that inputs similar to Shiplacoff’s “flick” input are typically programmed to trigger scrolling within an application. *See also* Terveen Deposition, 83:5-12 (“So in a particular context or state a flick could support scrolling.”).

96. For instance, a POSITA would have understood that, when a “calendar application” is active, such as the application being displayed in “full-screen mode” in Shiplacoff’s FIG. 2 (*see* Shiplacoff, ¶[0075]), substantially vertical gestures on the displayed calendar would conventionally be used for scrolling to other months, dates, and/or times in the calendar. This is evidenced in Anzures, which provides a FIG. 5A that—like Shiplacoff’s FIG. 2—depicts an interactive “calendar” being displayed across the full screen of a touchscreen device. Anzures, ¶[0169], FIG. 5A. For comparison, I have reproduced both Anzures’ FIG. 5A and Shiplacoff’s FIG. 2

below:

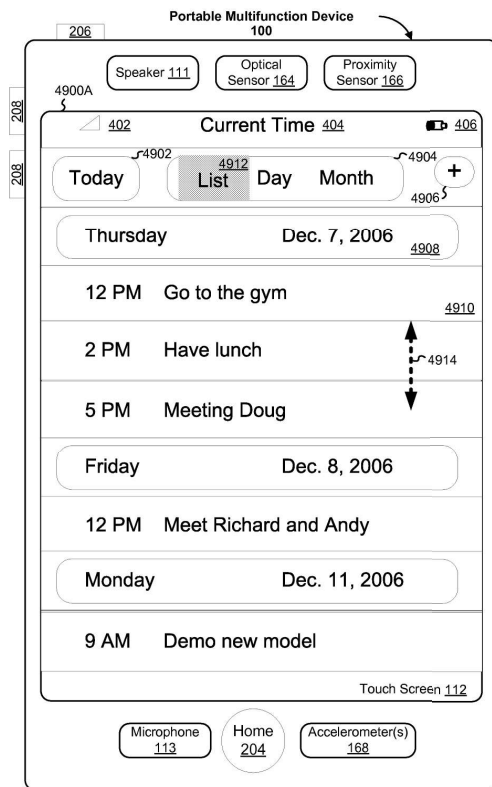


Figure 5A

Anzures, FIG. 5A

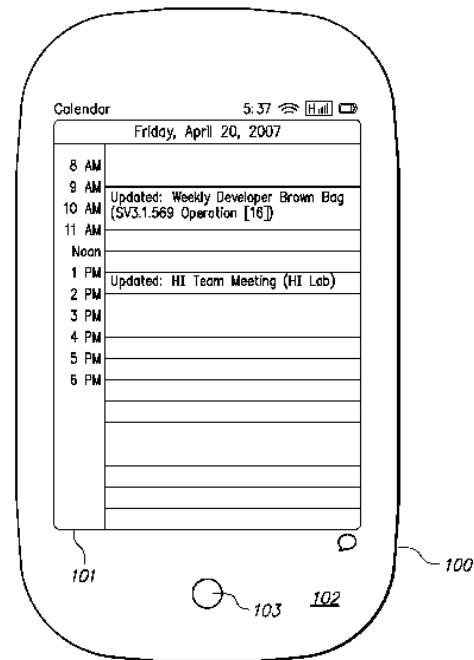


FIG. 2

Shiplacoff, FIG. 2

As shown, both figures depict a “calendar” having similar, manipulatable “on-screen elements,” including a list of “calendar entries” for various hours of a day. Anzures, ¶¶[0169]-[0180]; Shiplacoff, ¶[0076]. *Compare, e.g.*, Anzures, FIG. 5A (“Thursday Dec. 7, 2006”: “12 PM Go to the gym,” “2 PM Have lunch,” and “5 PM Meeting Doug”), *with* Shiplacoff, FIG. 2 (“Friday, April 20, 2007”: “9 AM Updated: Weekly Developer Brown Bag,” and “1 PM Updated: HI Team Meeting”). And as Anzures discloses, such “calendar” applications allow users to “scroll through the calendar by making a vertical finger swipe anywhere on the list calendar.” Anzures, ¶[0181]

(item numbering omitted); *see also id.*, ¶[0217] (explaining that, in a “day view” that displays only a single day’s events, the calendar is likewise “vertically scrollable to permit a user to see events earlier and later in the day” (item numbering omitted)), FIG. 7E (depicting an on-screen vertical gesture 7099 for scrolling); Shiplacoff, ¶[0076] (“The user interacts with the active activity (such as the calendar application shown in FIG. 2) by conventional user interface methods, including manipulating on-screen elements.... Such techniques are well known in the art.”).

97. As another example, a POSITA would have understood that, when an “email application” is active, such as the application being displayed in “full-screen mode” in Shiplacoff’s FIG. 5 (Shiplacoff, ¶[0090]), substantially vertical gestures would likewise typically be used for scrolling, including scrolling to other emails in the depicted inbox and scrolling through a selected email itself. This is evidenced in Jobs, which provides a FIG. 35A that—like Shiplacoff’s FIG. 5—depicts an interactive email “inbox” being displayed across the full screen of a touchscreen device. Jobs, ¶[0638], FIG. 35A. For comparison, I have reproduced both Jobs’ FIG. 35A and Shiplacoff’s FIG. 5 below:

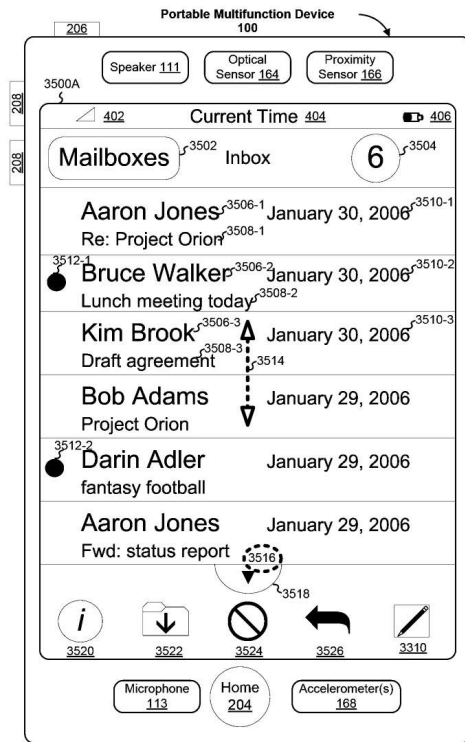


Figure 35A

Jobs, FIG. 35A

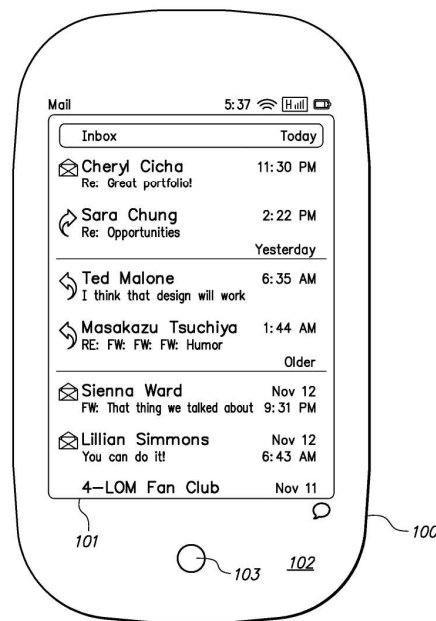


FIG. 5

Shiplacoff, FIG. 5

As shown at the top of both figures, each figure depicts an “Inbox” of various emails. *See also* Jobs, ¶¶[0638]-[0671]. Jobs explains that, when interacting with such email inboxes, users may “scroll through the emails using vertically upward and/or vertically downward gestures on the touch screen.” *Id.*, ¶[0672] (item numbering omitted). Indeed, Dr. Terveen acknowledges that “in th[e] email application that’s shown in [Shiplacoff’s] Figure 5, a user would expect to be able to scroll up and down in those messages.” Terveen Deposition, 41:15-22. Further, Jobs explains that, once an email is selected for display, the user may likewise scroll through the email’s contents using corresponding on-screen gestures. Jobs, ¶[0678].

98. In light of this conventional practice of using substantially vertical

gestures for scrolling within an application, which a POSITA would have been familiar with, as well as the disclosures of Shiplacoff and Duarte-992 that I discussed earlier, a POSITA would have avoided programming Shiplacoff's "flick" or other on-screen user inputs with similar attributes to trigger switching from "full-screen mode" to another mode so as to avoid any ambiguity about whether the input was instead intended for scrolling. *See* Terveen Deposition, 44:15-45:17 (agreeing that "within the same mode, a single gesture... has to have a single effect, like has to map to a single command, if you will. And that means by a single gesture it has to have something that's distinguishable to a user and the system"). Indeed, from my experience, and as confirmed by Duarte-992, mobile devices—particularly those with small screens—were recognized as being "relatively unforgiving in their interpretation of user input" and creating "difficulty activating the correct command or object." Duarte-992, ¶¶[0008]-[0009]. To avoid further difficulty, a POSITA would not have sought to use an input already used for scrolling in "full-screen mode," such as Shiplacoff's "flick," to trigger a switch from "full-screen mode" to another display mode or to the "launch screen."

99. Further, Shiplacoff not only teaches away from using its "flick" input to trigger switching from "full-screen mode" to a different display mode, but instead discloses that such a display change is triggered by a gesture or command beginning in an *off-screen* area. For example, Shiplacoff's ¶[0127] discloses that "switch[ing]

from full-screen mode” to another display mode is triggered by a “swipe up” gesture that “begin[s] *in gesture area 102*” (green, below) and “extend[s] onto screen 101” (red, below):

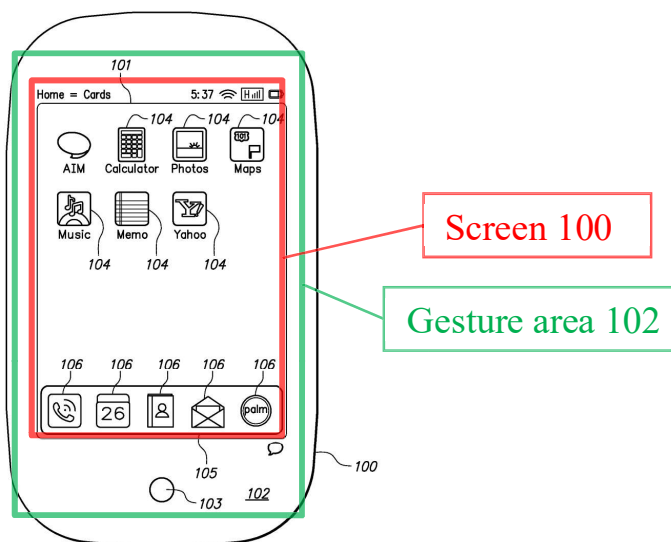


FIG. 1

’578 Patent, FIG. 1 (annotated), ¶¶[0127]; Duarte-992, ¶¶[0060], [0062]; Terveen Deposition, 35:18-36:16; *see also, e.g.*, Shiplacoff, ¶¶[0067] (“*[P]hysical button 103* can be used to toggle between full-screen mode and card mode[.]”), [0079] (“[T]he user can move cards 301 left and right by *dragging a finger along gesture area 102*[,]... even in full-screen mode[.]”).

100. By contrast, Shiplacoff’s ¶[0128] discloses that “switch[ing] *to* full-screen mode” from “card mode” is triggered by performing a “swipe down” gesture,” which “begin[s] *on screen 101* and extend[s] onto gesture area 102.” Shiplacoff, ¶[0128]. While Shiplacoff discloses that this “swipe down” gesture can

cause switching between *certain* modes, Shiplacoff does *not* disclose that its “swipe down” gesture triggers “switch[ing] *from* full-screen mode” to any other display mode or to the launch screen. *Id.*; Terveen Deposition, 37:14-38:13 (“[I]n Paragraph 128, there’s only -- there’s no examples given of switching from full-screen mode.”). In light of these disclosures, a POSITA would not have been motivated to modify Shiplacoff to use its “flick” input—which again is described as an input “performed *on/-/screen*”—to trigger switching from “full-screen mode” to another display mode or to the “launcher screen.” Duarte-992, ¶¶[0105], [0080], [0099]-[0100]; Shiplacoff, ¶¶[0060], [0112] (describing use of its “flick” input in “card mode” to close cards, which are “areas *of the screen* that can be moved and manipulated by the user”); *see also* Terveen Deposition, 46:12-47:13 (agreeing that, “in a hypothetical example where you have an identical gesture, it has to have a single effect. It can’t have an effect of doing something within the user interface mode and then switching to another mode”); Duarte-992, ¶[0008] (“[S]ome operations in connection with control of an electronic device are *not* particularly well suited to *direct* manipulation. These include operations that affect the *entire screen*, *application environment*, or the device itself.”).

3. Shiplacoff’s “Tap” Input Does Not Meet the Requirements of the Claimed “Second Input”

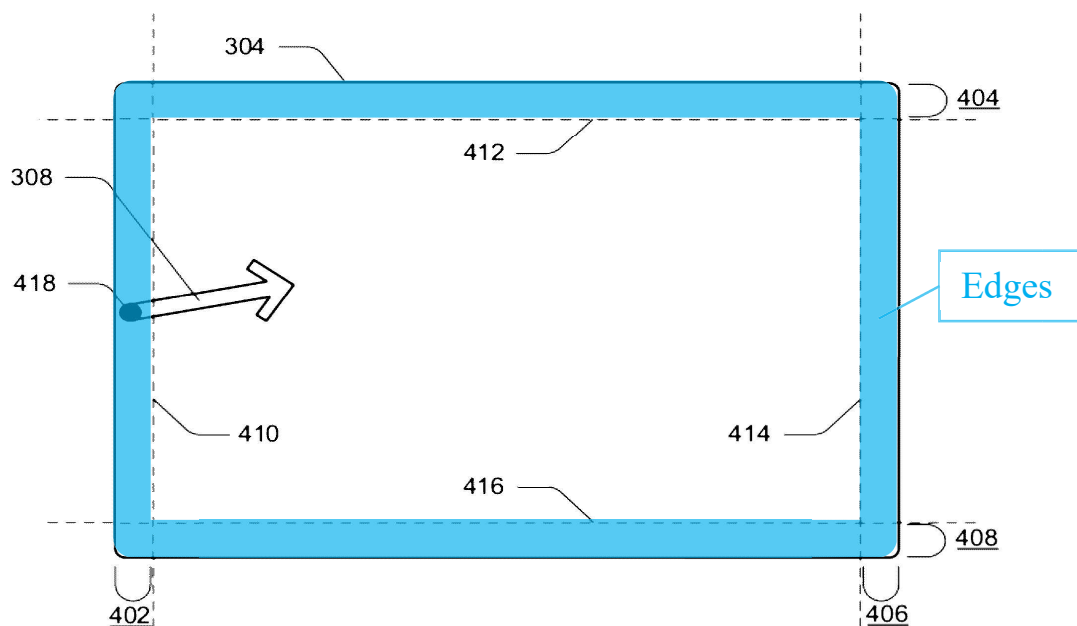
101. Independently, the Petition’s mapping of Shiplacoff’s “tap” input to “the second input” of Limitations [1k], [15g], and [27f] fails because a “tap” does

not “include[] movement across the display. As I explained above in Sections VI.B and VIII.B, Limitations [1i], [15e], [27d]—which recite detecting “*a second input* by a second contact that includes movement across the display in a direction”—provide the antecedent basis for “*the second input*” recited in Limitations [1j]-[1k], [15f]-[15g], and [27e]-[27f]. Accordingly, to satisfy Limitations [1k], [15g], and [27f], “*the second input*” must be “by a second contact that includes movement across the display in a direction.”

102. The Petition fails to identify any disclosure in Shiplacoff that a “tap” input meets all claimed requirements for the “second input.” While the Petition asserts that Shiplacoff’s “swipe up” gesture involves “movement across the display in a direction,” the Petition makes no such assertion with respect to Shiplacoff’s “tap” input. *See, e.g.*, Pet., 16 (describing only Shiplacoff’s “swipe up” gesture as involving “movement”). Nor does the Petition include any assertion that a POSITA would have understood a “tap” input to include “movement across the display.” *See, e.g., id.* Setting these failures aside, a POSITA would have understood the “tap” input described by Shiplacoff to be touchscreen input that does *not* include movement across the screen. This is consistent with Shiplacoff’s disclosure, which distinguishes between “tapping” on a card and “dragging” the card “from one position to another.” Shiplacoff, ¶[0104]. For these reasons, the Petition’s mapping of Shiplacoff’s “tap” input to Limitations [1k], [15g], and [27f] fails.

XI. THE PETITION’S GROUND 2 FAILS TO DEMONSTRATE THAT A POSITA WOULD HAVE BEEN MOTIVATED TO COMBINE SHIPLACOFF AND NAN AS PROPOSED

103. The Petition’s Ground 2 relies on a proposed modification to Shiplacoff’s device such that the device utilizes Nan’s “edge gesture.” *See* Pet., 23-25. However, the Petition fails to show that a POSITA would have sought to modify Shiplacoff’s device in such a way, as doing so would render Shiplacoff’s “gesture area” inoperable. Nan discloses using an “edge gesture” that includes “a start point... *at an edge*.” Nan, 4:35-36. Nan explains that “an edge” for an “edge gesture” is a “small” region of pixels that extends inward up to an “edge limit,” as shown in Nan’s FIG. 4, which I have reproduced and annotated below:



Id., 4:35-62, FIG. 4 (annotated). As shown, Nan’s FIG. 4 depicts four “edges” (blue), each of which lie between the “display limit” of an “immersive interface 304”

and an inner “edge limit.” *Id.*, 4:52-62.

104. Nan discloses that its device distinguishes between an “edge gesture” and other touch inputs by determining whether the “start point” of the touch event “is at an edge.” *Id.*, 4:35-5:2. For example, Nan discloses that the device “receiv[es] data indicating [X,Y] coordinates in pixels at which [a] gesture begins and compar[es] the first of these coordinates to those pixels contained within each edge.” *Id.*, 4:65-5:2 (item numbering omitted).

105. Although Nan discloses that an “edge” can be located beyond the display limit of the touchscreen “when an input mechanism is able to receive a gesture portion beyond the display or screen,” Nan’s disclosure regarding differentiating “edge gestures” from other touch inputs remain the same. *Id.*, 4:35-47. That is, Nan discloses that an “edge gesture” is identified by determining whether the starting point of a user input is within an “edge.” *Id.*, 4:35-5:2. Thus, in embodiments where the “edge” is off-screen, it is treated the same as an on-screen “edge” for purposes of detecting an “edge gesture.” *Id.*, 4:35-5:2.

106. Incorporating Nan’s edge gesture into Shiplacoff’s device would render Shiplacoff’s off-screen “gesture area” inoperable. As I explained above in Section X.B, Shiplacoff discloses that its device distinguishes between gestures and other touch inputs based on whether the touch input includes contact with this “gesture area,” such as a “cross over” between the gesture area and the screen. If Shiplacoff

were modified to utilize Nan’s “edge gesture” as proposed by the Petition, on-screen and off-screen touch inputs would no longer be differentiated from one another. Rather, in accordance with Nan’s disclosures, whether a touch input occurs off-screen and on-screen would not be a basis for differentiation between inputs. *See* Nan, 4:35-5:2. Instead, an “edge gesture” would be distinguished from other touch inputs based on whether the touch input starts in an “edge,” regardless of whether the “edge” is on-screen or off-screen.

107. A POSITA would ***not*** have been motivated to render Shiplacoff’s “gesture area” inoperable for its intended purpose and thereby negate the benefits associated with it, such as “expand[ing] the available space and the vocabulary of gestures.” *Compare id.*, with Duarte-992, ¶[0072]. Indeed, Duarte-992—which Shiplacoff incorporates by reference—teaches away from using gestures such as Nan’s “edge gesture.” Duarte-992 discloses that, while an area comparable to Nan’s “edge”—i.e., a “gap between [the] gesture area and screen”—may be present, the device will “simply ***ignore*** the gap when interpreting touch commands that cross over from gesture area 102 to touch-sensitive screen 101, and vice versa.” Duarte-992, ¶[0060] (item numbering omitted); Nan, 4:35-62. In light of these disclosures, a POSITA would not have been motivated to modify Shiplacoff’s device to utilize this “gap” or “edge” of Shiplacoff’s touchscreen for differentiating user inputs, and thus would not have found Nan’s “edge gesture” suitable for use in Shiplacoff’s

device.

XII. THE PETITION FAILS TO ESTABLISH THAT SHIPLACOFF’S “TAP OR FLICK” INPUT DISCLOSES DEPENDENT CLAIMS 2-3, 16-17, AND 28-29

108. As I explained above in Section VIII.B, Dependent Claims 2-3, 16-17, and 28-29 recite further constraints on “detecting *the* second input” of Limitations [1i]-[1k], [15e]-[15g], and [27d]-[27f].

109. With respect to Dependent Claims 2, 16, and 28, the Petition’s arguments are limited to Shiplacoff’s “swipe up” gesture and Nan’s “edge gesture,” which the Petition maps alternatively (in both Grounds 1 and 2) to “the second input” recited of Limitations [1i]-[1j], [15e]-[15f], and [27d]-[27e]. Pet., 25-26 (“[I]t would have been obvious to a POSITA that, at least part of the ‘swipe up’ gesture (notably the part that starts at the edge of screen 101) would have included ‘movement that starts at an edge of the display.’” (emphasis omitted)), 35, 40, 47, 49. Setting aside that the Petition’s mapping of Shiplacoff’s “tap or flick” input to “the second input” in Limitations [1k], [15g], and [27f] fails for the reasons I explained above in Section X.C, I find nothing in the Petition that explains how Shiplacoff’s “tap or flick” input satisfies the additional requirements of Dependent Claims 2, 16, and 28. *See* Pet., 25-26, 35, 40, 47, 49.

110. Nonetheless, had the Petition attempted to map Shiplacoff’s “tap or flick” input to Dependent Claims 2, 16, and 28, such a mapping would have failed

because Shiplacoff does not disclose or suggest a “tap or flick” input that “includes movement that starts at an edge of the display.” Again, as I explained above in Sections X.C.1-X.C.2, Shiplacoff’s “flick” input is disclosed as being “performed *on/-/screen*” for “*direct* manipulation of” on-screen objects such as “cards.” The “flick” input thus starts “*on/-/screen*” and not “at an edge of the display” as claimed. Further, as I explained above in Section X.C.3, a “tap” input contains no “movement” and thus equally fails to meet the claimed “movement that starts at an edge of the display.”

111. The Petition’s mapping with respect to Dependent Claims 3, 17, and 29 are similarly deficient. *See* Pet., 26-27. The Petition’s arguments for Claim 3 consist entirely of a cross-reference to the Petition’s contentions for Limitation [1i] and Dependent Claim 2, which again are limited to allegations that Shiplacoff’s “swipe up” gesture and, alternatively, Nan’s “edge gesture” satisfy those limitations. *Id.*; *see also id.*, 34, 40, 47, 49. I again find nothing in the Petition explaining how Shiplacoff’s “tap or flick” input satisfies the additional requirements of Dependent Claims 3, 17, and 29.

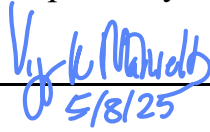
XIII. CONCLUSION

112. All statements made herein of my own knowledge are true, all statements made herein on information and belief are believed to be true, and these statements were made with the knowledge that willful false statement and the like

are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Respectfully submitted,

Dated: May 8, 2025



Dr. Vijay Madisetti