

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.,

Petitioner

v.

SMITH INTERFACE TECHNOLOGIES, LLC

Patent Owner

IPR2024-01088

Patent No. 10,656,758

DECLARATION OF DR. VIJAY MADISETTI

TABLE OF CONTENTS

I. Introduction.....1

II. Professional Background.....1

III. Compensation 7

IV. Legal Consideration.....8

 A. Obviousness.....8

 B. Claim Interpretation 9

V. Overview of Task & Basis for Opinions 10

VI. Level of Ordinary Skill in the Art 11

VII. Overview of the ’758 Patent.....12

 A. Multi-Part Pattern-Based Gestures.....12

 B. Independent Claims 1, 16, and 84.....18

VIII. Claim Interpretation 20

 A. Logic Based Limitations 20

 B. “The Duration Of The First Single-Finger Touch Input” 21

 C. Touch Continuity between Limitations [1.k] and [1.l] 25

 D. User Interface Element.....27

IX. Overview of the Prior Art.....29

A.	U.S. Patent Application Publication No. 2011/0252372 to Chaudhri (“Chaudhri”).....	29
B.	U.S. Patent Application Publication No. 2004/0051726 to Martyn (“Martyn”).....	31
X.	Petitioner fails to properly map “the first application icon.”	35
A.	Petitioner Fails to Consistently Map “The First Application Icon”	35
B.	Petitioner Fails to Identify Moving “The First Application Icon” Based on “The First Single-Finger Touch Input”	38
1.	Petitioner’s Mapping to Determining A Different Touch Input After “The First Single-Finger Touch Input Concludes” Meets Movement Criteria Does Not Meet the Claim Language	38
2.	Petitioner Proposes Further Modifying Without a Real Motivation	40
XI.	A POSITA would not have taken Petitioner’s desired calling method for calling Martyn’s Snapshot	44
A.	Petitioner Contorts the References to Reach the Purported Combination ..	44
B.	A POSITA would not have been motivated to choose a specific method of calling Martyn’s SnapShot.....	46
C.	Chaudhri Already Discloses A Calling Method for A Context Menu.....	53

D. The Long Touch Gesture Is already Assigned to a Different Gesture
Teaches Away From Assigning That Gesture To Calling The App Snapshot 59

E. Numerous Available User Gestures That Are Consistent, Rather Than
Conflict With Chaudhri and Martyn’s Teachings, Further Teach Away From
Selecting a Gesture That Creates A “Conflict” 60

1. Consider Martyn’s “press and hold of the select function when an
application name is highlighted” as disclosed avoids conflict 60

2. Martyn’s other methods of calling up the App Snapshot disclose further
options avoiding conflict with Chaudhri 63

F. A Self-Induced Conflict is the Only Motivation for Adding Griffin.....66

XII. Petitioner fails to properly map the required claim elements in multiple
dependent claims.....67

A. [21] and [22]: Content Indication.....68

B. [35]: Multiple Moved Apps.....70

C. [47]: Interactive User Element 71

D. [57]: Dynamically Generated Content 72

XIII. CONCLUSION 73

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. 10,656,758 (“the ’758 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 EX2006. 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, Berkeley and the IEEE/ACM Ira M. Kay Memorial Paper Prize in 1989.

5. 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, data analytics, wireless communications, messaging protocols, microprocessor architecture, hardware, RF, cellular networks, computer engineering, digital signal processing, 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 in particular. 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.

6. 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.

7. 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 2000-2001 timeframe I designed three GSM multiband mobile phones for a leading telecom equipment manufacturer in Asia, and in the

2002-2007 timeframe, 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.

8. 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>).

9. 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.

10. 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, and am listed as an inventor on several patents related to wireless messaging. Many of my patents also 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.

11. Additionally, I have extensive experience analyzing, designing, and testing systems based on 3GPP Technical Specifications, including specifications describing WCDMA and HSDPA technologies. I have been active in the area of location-based services and wireless localization techniques since the mid-1990s, and have authored several papers on location-based services, including, Vijay K. Madisetti et al., Mobile Fleet Application Using SOAP and System on Devices (SyD) Middleware Technologies, Communications, Internet, and Information Technology (2002) at 426-31.

12. 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.

13. 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. In 1989, I was recognized with the Ira Kay IEEE/ACM Best Paper Award for Best Paper presented at the IEEE Annual Simulation Symposium.

III. Compensation

14. 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 Consideration

15. I am not an attorney. My understanding of the law is based on information provided by counsel for Smith Interface Technologies, LLC.

A. Obviousness

16. I understand that a claimed invention is obvious and, therefore, not patentable if the subject matter claimed would have been obvious to a person of ordinary skill in the art at the time that the invention of the '758 Patent (a "POSITA"), which I have been asked to treat as August 5, 2011.

17. I understand that a claim can be obvious in view of a single prior art reference (e.g., via modification of that prior art reference) or multiple prior art references (e.g., via a combination of two or more prior art references), if such a modification or combination was within the skill of a POSITA. I understand that there must be some articulated reasoning with some rational underpinning to support a conclusion of obviousness. I also understand that to establish a finding of obviousness one must show that one of ordinary skill in the art would have had a motivation to combine the prior art references to produce the claimed invention and a reasonable expectation that the combination would be successful.

18. I further understand that exemplary rationales that may support a conclusion of obviousness include: (1) simply arranging old elements in a way in which each element performs the same function it was known to perform, and the arrangement yields expected results, (2) merely substituting one element for another known element in the field, if the substitution yields no more than a predictable result, (3) combining elements in a way that was “obvious to try” because of a design need or market pressure, where there was a finite number of identified, predictable solutions, (4) that design incentives or other market forces in a field would have prompted variations in a work that were predictable to a person of ordinary skill in the art, and (5) that there was some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify or combine prior art references to arrive at the claimed invention.

B. 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), 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 '758 Patent. I have been asked to provide opinions related to certain issues from the perspective of a person of ordinary skill, having knowledge of the relevant art, as of August 5, 2011, 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 VI below.

21. More particularly, I have been informed that the Patent Trial and Appeal Board (the “Board”) will decide whether the Petition establishes by a preponderance of the evidence that at least one claim of the '758 Patent is invalid. I have been asked to focus my analysis on the Petition’s proposed invalidity theories related to the challenged claims of the '758 Patent.

22. 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 documents listed in the Exhibit List included with the Petition, the deposition of Dr. Cockburn, Dr. Cockburn, and any other items that I reference in my below analysis.

23. 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. Level of Ordinary Skill in the Art

24. I understand that a patent must be analyzed from the perspective of a person of ordinary skill in the art (“POSITA”) as of the time of invention of the patent, which is typically considered to be the patent’s earliest filing date.

25. I agree with the level of ordinary skill in the art adopted by the Board for the purposes of the Institution Decision. *See* Paper 7, 8-9. Specifically, a POSITA:

“would have had ‘a bachelor’s degree in computer science, computer engineering, or a comparable field and at least two years of professional experience working with graphical user interfaces and applications for computers.’”

Id., 8.

26. I also agree with the Board’s rejection of two other parts of the Petition’s level of ordinary skill. First, the phrase “[a]dditional years of experience could substitute for an advanced-level degree (and vice versa)” (Pet., 7) is inapplicable where the base degree is a bachelor’s degree as here and was likely a simple copy and paste error. Dr. Cockburn confirmed that deleting “additional years of experience could substitute for the advanced-level degree” from his definition cited in the Petition and would make “more sense.” *See* EX2007, 9:16-23 (“[Q.] Would you agree that the same is true here that, the definition you’ve given in

[EX1003, ¶39] makes more sense if we delete the end of the last sentence which is and additional years of experience could substitute for the advanced level degree?

A. I think that’s fair. We could terminate the sentence at that final comma.”).

27. Second, I agree with the Board that Petitioner’s additional statement that a POSITA “would have possessed extensive background knowledge and understanding of the pertinent technologies” listing specific areas (Pet., 7) should not be adopted. *See* Paper 7, 9. This additional requirement is either redundant with the level of ordinary skill articulated above, or inconsistent to the extent it seeks to impose a level of ordinary skill beyond “a bachelor’s degree ... and at least two years of professional experience.” The background knowledge and understanding of touch screen mobile devices, menus in computer graphical user interfaces, and duration-based touch gestures technologies would be a part of the bachelor’s degree and at least two years of experience considering the course work required by those degrees and the exposure to those features while in the industry.

VII. Overview of the ’758 Patent

A. Multi-Part Pattern-Based Gestures

28. At the time of the ’758 Patent’s December 2011 priority date, touch screen mobile devices were becoming increasingly capable of more complex tasks that required more control than possible with traditional, basic input methods. As

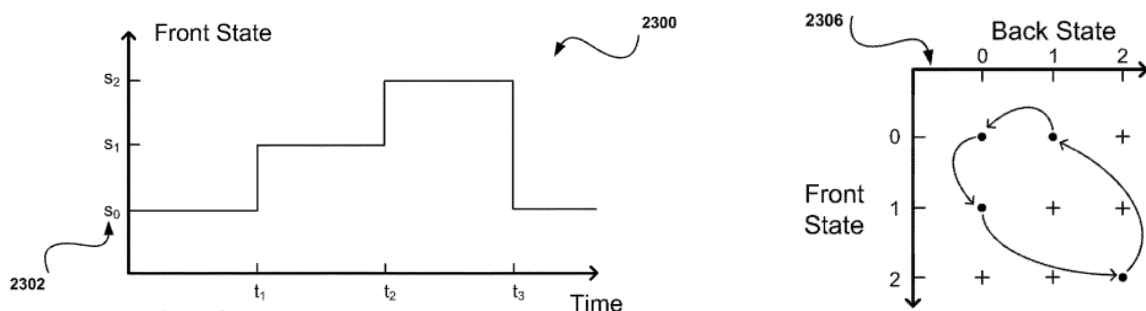
more capable and precise sensors were being placed in mobile devices, the corresponding options for designers were increasing.

29. Additionally, advancements in cellular connectivity speed enabled more complex tasks on touch screen mobile devices such as gaming and more detailed remote office tasks. The introduction of 4G networks offered up to ten times faster speeds. Greater cellular speeds not only enabled enhanced multi-player gaming but also enabled users on the go to more readily access games, which in turn increased demand for mobile games. In person interactions with mobile phone users partly drove this demand. For example, when a user would show another user a game they had found, the second person would desire immediate access on their own mobile device. This immediate access was facilitated by greater cellular speeds leading to higher adoption rates. Remote office work performed on mobile devices also increased with increasing cellular speeds. Increased cellular speeds enabled easier document access (e.g. presentations) on mobile devices. This in turn increased demand for modifying those documents on the go. The spread of mobile gaming and games as well as increased out of office work possibilities led to increasing capabilities for more complex tasks that required advancements over basic input methods.

30. Further, limited screen real estate meant that input and output techniques used for more complex tasks on desktop computers could not be used.

The '758 Patent addresses this shortcoming by enabling users to operate these new powerful mobile devices, which had less real estate, and their feature-rich applications through implementation of multi-part gestures where users can take successive actions and receive feedback at each stage.

31. The '758 Patent describes these multi-part, multi-action gestures as “pattern-based gestures” that “may be recognized in the form of an ordered sequence” by, for example, “using a decision tree.” EX1001, 75:37-39; EX2008, [00581]. The “pattern-based gesture” may be defined as “a path through a phase space of one or more dimensions.” EX1001, 73:39-41; EX2008, [00564]. Exemplary “pattern-based gesture[s] 2302” are “described by a phase space path which represents the time evolution of the touch state associated with a contact point” on the touch screen. EX1001, 73:53-56; EX2008, [00566].

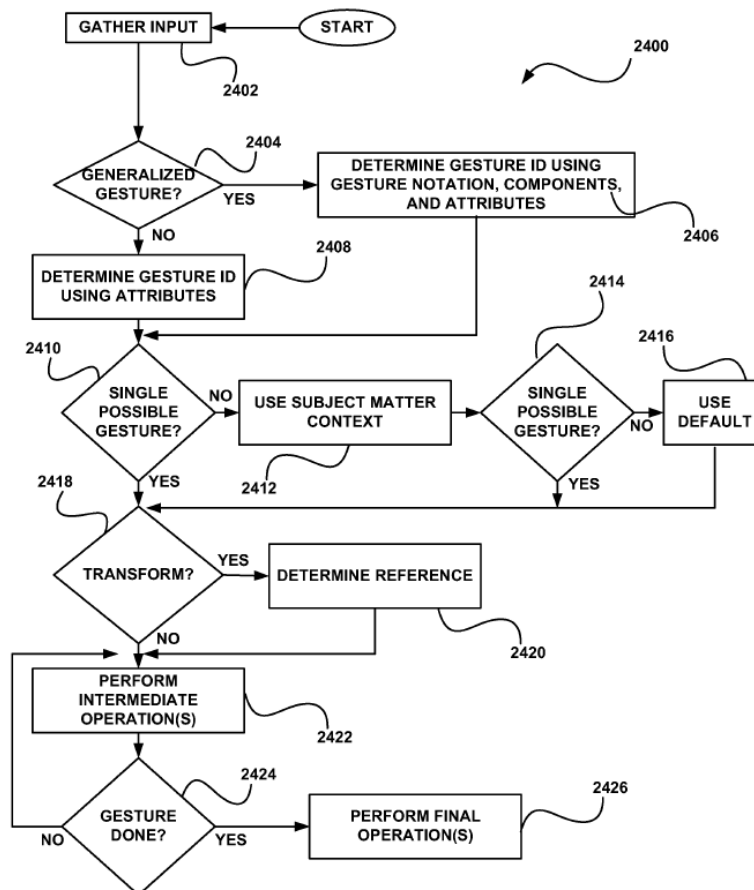


EX1001, FIG. 23; EX2008, FIG. 28. Each state, and transitions between states, may be defined by factors including “contact point changes” and “the time spent on each leg of the path through phase space.” EX1001, 73:59-74:19; EX2008, [00567]-

[00570]. Consistent with the “decision tree” aspect of these pattern-based gestures, multiple patterns may share the same starting (and intermediate) points. EX1001, 75:37-39; EX2008, [00581]. Consistent internal device logic is important to keep a user engaged and appropriately aware of the response for a desired task. A user wishing to execute a final action contingent on a previous device state would be unsuccessful if the device did not predictably follow a decision tree. Predictability in starting and intermediate points is enabled by decision tree logic.

32. To detect these pattern-based gestures, the '918 Provisional, which the '758 Patent incorporates by reference (EX1001, 1:24-26), discloses techniques for “gesture differentiation,” which is distinguished from simply detecting a “basic user interaction (i.e., button press, basic application interaction, etc.).” EX2008, [00498], [00502]. The process begins by “determine[ing] whether there is any meaningful user input,” including applying a “threshold above which user interaction may rise before it may be deemed meaningful.” *Id.*, [00501]-[00502]. Detecting the gesture may involve “gesture subject matter context,” including “objects displayed at or near the location of the gesture.” *Id.*, [00512]. For example, the starting point may be the beginning of a touch contact at a particular location of a specific object on the touch screen. *Id.* A touch-capable device offers a user the ability to input at a desired location on the touch screen. The ability for a user to pick from a variety of locations on a touch-capable device requires device logic to start a decision tree with a

detected user touch contact at a specific location. This location often corresponds to a desired object like an application. This location could also refer to other objects that a user would recognize that interacting with could lead to a desired result.



EX2007, FIG. 24.

33. Again, as indicated by the flow chart, multiple complete full patterns may share the same starting (and intermediate) points. From the flow chart, the start of a pattern begins with gathering input at block 2402. Further blocks, for example block 2406, describe gesture notation, components, and attributes. This reveals that the input gathering at one location could be at the same location based on later

defined features of that input. Furthermore, the steps of the flow chart illustrate that a gesture pattern involves multiple steps that may share the same starting point as well as intermediate points. For example, a different gestures or input type at the same location would thus share the same starting point. Other examples consistent with the '758 patent could include a common starting point in a pressure level. In both instances, the logic of the device analyzes a parameter associated with user contact. The same pressure level again could be the same starting point. Consistent with the multi-part nature, the device may perform “intermediate operations associated with the identified gesture.” *Id.*, [00521]. “[E]ach execution of intermediate gesture operations may trigger a display update (i.e. rendering, etc.).” *Id.*, [00524]. One purpose of these intermediate gestures is so “a user may receive feedback on their gesture before terminating the interaction.” *Id.*, [00522].

34. A “gesture may be ended by ending one or more user inputs” or “by performing a different gesture” at which time “the final operations associated with the identified gesture are performed.” *Id.*, [00526]-[00528]. “For example, in one embodiment, a gesture which triggers the display of a pop-up menu and allows a user to cycle through menu items may have as a final operation the selection of the current menu item and removal of the pop-up menu.” *Id.*, [00529].

B. Independent Claims 1, 16, and 84

35. Petitioner challenges the three independent claims of the '758 Patent: Claims 1, 63, and 124 (“Challenged Independent Claims”). Claim 1 is directed to an “electronic device.” EX1001, 97:29. Claim 63 is directed to a method. EX1001, 108:41. Claim 124 is directed to a “non-transitory computer readable storage medium.” EX1001, 114:60. Claims 1, 63, and 124 are substantially similar.

36. The Challenged Independent Claims recite logic to process user inputs on a touch screen to recognize a set of gesture components that begin from a common starting point and perform operations and/or provide feedback upon recognizing various aspects of a gesture. Consistent with the above description, certain inputs could be confined to a single type of input or impose certain requirements to result in a desired output or operation. The logic to process user inputs could then be defined accordingly as has been done for the claims of the '758 patent. Claims 1, 63, and 124 each recite “an application launching user interface that includes a plurality of application icons,” detecting a first single-finger touch input” “at a location on the touch-sensitive surface that corresponds to a[n] ... application icon,” and “determining a response based on evaluating the” “duration” or “movement” of the “touch input against” various criteria. Limitations [1.f-g], [63.b-d], [124.a-c].

37. This sort of decision tree logic is described both in the '758 Patent and in the '918 Provisional, as explained in above in ¶¶28-34.

38. The Challenged Independent Claims recite, in Limitations [1.i-l], [63.e-h], [124.d-g], the following logic and corresponding device functionality:

- “[in accordance with] / [based on] a determination that ... the duration of the first single-finger touch input is evaluated to be less than a first time threshold, launching the first application,”
- “[in accordance with] / [based on] a determination that the first single-finger touch input ... the duration of the first single-finger touch input is evaluated to be greater than the first time threshold, displaying one or more action option objects associated with the first application without launching the first application,”
- “[in accordance with] / [based on] a determination that the first single-finger touch input ... the duration of the first single-finger touch input is evaluated to be greater than a second time threshold that is greater than the first time threshold, performing an operation ... , and”
- “[in accordance with] / [based on] a determination that the first single-finger touch input meets the one or more movement criteria, moving the first application icon in a foreground virtual display layer so that the first application icon appears to float above a background virtual display layer.”

VIII. Claim Interpretation

A. Logic Based Limitations

39. Limitations [1.i-l], [63.e-h], [124.d-g] collectively recite four actions to be performed “[in accordance with] / [based on]” a “determination that the first single-finger touch meets” four recited “criteria.” These four Limitations collectively describe logical functions/steps to process instances of user input on a touch screen that begin from a common starting point—the start of “the first single-finger touch input.”

40. A careful read of the claims reveals a specific word choice that is narrow. The claim uses the term “single-finger touch input” rather than something broader like a gesture. The word choice confines the device logic to this specific type of input. A POSITA would arrive at this conclusion by understanding the notion of a “decision-tree” and device logic for detecting multi-part pattern-based gestures as described in connection with the states described in Figure 23 of the ’758 Patent and the flow diagram in Figure 24 of the incorporated-by-reference ’918 Provisional as discussed above in ¶¶28-34.

41. Reading the claims and the connections between each element reveals that the logic of a device must be able to detect what is described in each element. The claims as written do not suggest that a device could be programmed to detect one criteria and not another. The fact that some criteria could not be simultaneously

met does not change that the logic of the device must be able to detect all of the claimed criteria. Furthermore, when designing a user interface with multiple criteria, the logic for satisfying those criteria would still be present even if not all of those criteria are simultaneously or frequently met. The form of the claim does not change this analysis. To appreciate the claimed invention from the appropriate perspective of the logic of a device, understanding multiple conditional steps is required. For example, Claim 63 includes multiple conditional steps that specify different actions based on the evaluation of the touch input against various criteria. Each condition leads to a different action, and understanding the claim fully requires evaluating all possible conditions and their corresponding actions.

B. “The Duration Of The First Single-Finger Touch Input”

42. Each Challenged Independent Claim recites “detecting a first single-finger touch input” and then recites multiple limitations each referencing different “determination[s] that the first single-finger touch input meets” a criteria, including three limitations that reference “the duration of the first single-finger touch input.” Limitations [1.g-l], [63.c-h], [124.b-g]. The claims do so in the context of describing programming logic that will receive and act upon numerous instances of user input in accordance with that logic. *See* Limitations [1.e] (“the one or more programs including instructions for:”), [63.a] (“at an electronic device”), [124.pre] (“the one or more programs comprising instructions which ... cause the electronic device to”).

Again the specific word choice of the claim conveys a particular example of logic.

The freedom of designers to choose among different options leads designers to choose particular implementation of criteria and actions. In this context, the claims outline the desired input of a first single-finger touch and the duration associated with it. A duration could not be of varying degrees without it being the same single-finger touch input. Accordingly, the claims lay out that the logic of a device must detect this single-finger touch input.

43. The antecedent bases in the claims anchor the reading of the claims. The antecedent bases “the duration of” and “the first single-finger touch input” indicate that evaluating the preconditions recited in each Limitation requires evaluating instances of user input beginning at a common starting point as discussed above in ¶¶28-34.

44. The claim language not only specifies a *single*-finger touch input but also specifies the location of that single-finger touch input at a location of a first application icon. As described in Limitations [1.g], [63.c], [124.b] “detecting a first single-finger touch input that includes detecting the first single-finger touch input at a location on the touch-sensitive surface that corresponds to a first application icon of the plurality of application icons.” This again represents the confining of device logic by the claims. By confining the logic of the device the claim is not met if a single-finger touch input is detected at other locations on a touch-screen device.

45. The claim limitations lay out that a duration must be evaluated to trigger certain criteria. For example, in Limitations [1.i-k] the claims require “the duration of the first single-finger touch input is evaluated to be less than a first time threshold,” “the duration of the first single-finger touch input is evaluated to be greater than the first time threshold,” and “the duration of the first single-finger touch input is evaluated to be greater than a second time threshold that is greater than the first time threshold.” These Limitations describe three different durations and the logic of the device must be able to detect each duration to satisfy the claims. Logic only detecting a duration less than a first time threshold, is not logic that is also capable of detecting a duration greater than a second time threshold. Likewise, if the duration does not start from the same starting point the logic of the claims cannot be followed because a duration of the same single finger touch input could not be compared with a threshold. In other words, if the starting point is not the same, there would not be logic for “evaluating *a duration of the first single-finger touch* input against at least one of” several criteria.

46. The specific claim language opts for a single-finger touch input. This confines the logic as described above. A *multi*-finger touch input would be different and not meet the required claim language.

47. As previously discussed, this understanding is consistent with the “decision-tree” and detection logic for detecting multi-part pattern-based gestures as

described in connection with Figure 23 of the '758 Patent and Figure 24 of the incorporated-by-reference '918 Provisional.

48. It is also consist with the fact that some, but not all, of the Limitations recite mutually-exclusive preconditions such that requiring all preconditions to be met in response to a single instance of user input is contrary to the plain claim language.

49. For example, consider Limitations [1.i] and [1.j]. Both of the following cannot happen simultaneously: “***launching the first application***” “in accordance with a determination that the first single-finger touch input meets the one or more application-launch criteria that is met when the duration of the first single-finger touch input is evaluated to ***be less than a first time threshold***” and “displaying one or more action option objects associated with the first application ***without launching the first application***” “in accordance with a determination that the first single-finger touch input meets the one or more action-option-display criteria that is met when the duration of the first single-finger touch input is ***evaluated to be greater than the first time threshold***.” In other words a touch cannot both be longer and shorter than a first time threshold, nevertheless device logic must ***be able to detect*** the duration for the respective criteria.

C. Touch Continuity between Limitations [1.k] and [1.l]

50. Petitioner identifies a claim scope issue, but does not take a position, as to whether the Challenged Claims should be “interpreted to require a single continuous touch between Elements [1.k] and [1.l].” Pet., 38. As an initial matter, the ’758 Patent’s disclosure of multi-part gestures includes multi-part gestures that do **not** involve a single continuation touch across all parts as described above in ¶¶28-34. This is also described in the 918 provisional. *See* EX2008, [00529] (describing a multi-part “gesture” that “triggers the display of a pop-up menu,” “allows a user to cycle through menu items,” and “may have as a final operation the selection of the current menu item and removal of the pop-up menu”).

51. However, the specific claim language of the Challenged Claims requires a continuous touch as described in ¶¶28-34 and 35-38. The specific word choice of the claims cannot be discarded as that is the desired logic of the device. The claims could have been written in a different manner and would be describing a device with different logic for detecting touch input. Device logic is what enables different inputs to reach different results. Multiple options exist for designing user input and hence the series and requirements for that input are paramount.

52. Specifically, Limitation [1.g] recites “detecting **a** first single-finger touch input” “at a location ... that corresponds to a first application icon,” and subsequent Limitations [1.k] and [1.l] each refer to “**the first** single finger touch.”

Limitation [1.k] further refers to “the duration of the first single finger touch,” referring to the elapsed time since the initial touch detected in Limitation [1.g]. To consider the claims otherwise would render the duration detection meaningless. The first single-finger touch input initiates duration tracking and it is not until the finger is removed from the touch screen that the duration ends. For example, Limitation [1.i] requires “launching the first application” “when the duration of the first single-finger touch input is evaluated to be *less* than a first time threshold.” A single-finger touch input could not be less than the first time threshold without the duration of the touch ceasing. Further, the context of the Challenged Claims, including the use of the word “touch” and identifying the start and duration of that touch, as well as the use of “the first single finger touch” in each limitation connote to a POSITA that each refers to a touch that has not concluded until lift-off.

53. The language in the Challenged Independent Claims assessing movement criteria for “the first single finger touch” stands in contrast to the language of other family members, such as Claim 1 of U.S. Patent No. 10,671,212, which recites “a touch” and “the touch” for several claim elements, but then recites separate subsequent detection of “a slide or swipe gesture.” Thus, the Challenged Independent Claims could have referred to “a” different gesture to refer to a different touch but they do not.

54. Appreciating these language differences reflects the choice of logic implementation. Since a variety of user inputs are available, verbiage differences are important to understanding the difference in implementation at a logic level.

D. User Interface Element

55. The plain and ordinary meaning of a “user interface element” is an element of a device that a user *must be able* to interact with. Considering otherwise would not give the appropriate limited meaning to a user interface element. A device may need to communicate multiple types of information to a user without each of those elements being interactive. For example, text (e.g. application or folder names) on a device is not necessarily interactive without a user interface element or other underlying interactive element enabled. Likewise color differentiation or other visual cues may not be interactive. This stands in contrast to an element that a user is able to interact with to bring about a desired goal. A user interface is the medium between a user and the underlying logic of the device. Accordingly, a visual cue on a device that does not enable interaction with a user does not comport with the plain and ordinary meaning of a user element.

56. The specification of the ’758 Patent also confirms this plain and ordinary meaning of user interface element. The specification of the ’758 Patent discusses “context determination zone[s].” EX1001, 28:60-67, 29:27-36. A “context determination zone refers to the area that is examined while determining the context

of a selection.” EX1001, 28:43-44. Further “a context determination zone may be chosen such that it is smaller than any user interface elements which may be detected.” EX1001, 28:62-64. Similarly, a “determination” of a “contact point” related to a “context determination zone” of a “UI control widget” “may be performed using the bounding boxes associated with user interface elements.” EX1001, 29:27-32. These context determination zones determine the contours of what a device could sense and hence a user could interact with.

57. The specification of the ’758 Patent also discusses “gestures” in relation to user interface elements. EX1001, 56:13-17, 66:3-13. The ’758 Patent explains that “[i]n various embodiments, a tipping pressure gesture may be used to interact with user interface elements.” EX1001, 66:3-13. A gesture is a form of input for a device and input requires interaction with a device.

58. The term “tipping pressure gesture” used in the ’758 Patent refers to a change in the location of an amount of pressure exerted by a user’s finger on a touch screen device. *See e.g.* EX1001, 66:17-21 (“While performing a tipping pressure gesture, if a user allows the contact pressure differential to switch sign (i.e. low pressure contact point becomes the high pressure contact point, and visa versa), a rocker pressure gesture may be recognized by the device.”).

IX. Overview of the Prior Art**A. U.S. Patent Application Publication No. 2011/0252372 to Chaudhri (“Chaudhri”)**

59. Chaudhri describes “electronic devices with touch-sensitive surfaces, including but not limited to electronic devices with touch-sensitive surfaces for managing folders.” EX1004, [0003].

60. Chaudhri describes a device that has numerous input methods. EX1004, [0085], [0087], [0093], [0095], [0097], [0101]-[0102], [0152], [0164], [0173]-[0174], [0211], [0218], [0380].

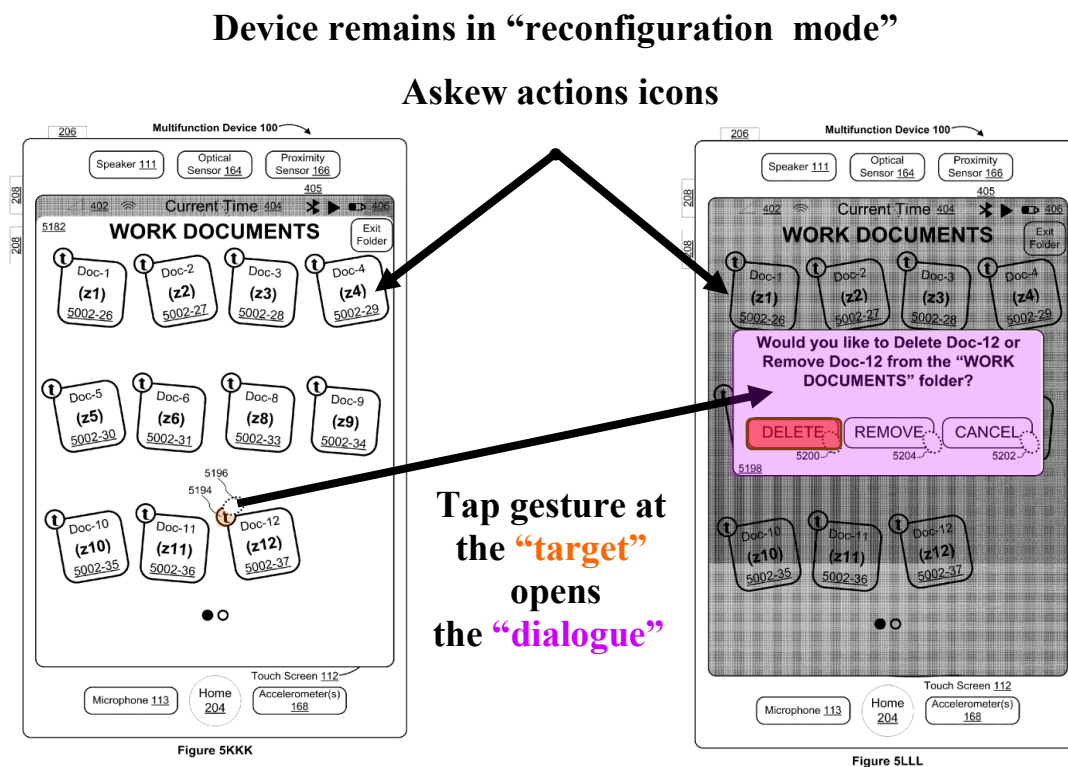
61. Chaudhri describes “input control devices [such as] physical buttons (e.g. push buttons, rocker buttons, etc.),” a “physical or virtual wheel (e.g. a click wheel) as an input control device,” an “optical sensor controller 158 in [an] I/O subsystem,” and that an “accelerometer 168 may be coupled to an input controller 160 in [an] I/O subsystem.” EX1004, [0085], [0093], [0095], [0097]. Physical buttons could be located on the side top or front of the device as shown in Chaudhri’s figures. EX1004, FIGS. 5A-5PPP. Button signals to a user a path for providing physical input by being permanently visible to a user. Likewise for a physical click wheel. A virtual click wheel would appear on the screen of Chaudhri’s device. Each of these offer a variety of avenues for a user to interact with Chaudhri’s device.

62. Chaudhri also describes “verbal input for activation or deactivation of some functions through microphone 113.” EX1004, [0176]. For example, a series of words or phrases could be used to start and stop certain functions.

63. Chaudhri also discloses two modes: a “normal operation mode” and a “user interface reconfiguration mode.” EX1004, [0217-0218] In the “normal mode, tapping on an application will launch the application. EX1004, [0217]. When in “user interface reconfiguration mode,” a user is able to rearrange applications and can reach a “pop-up dialogue” by touching a separate “object modification target.” EX1004, [0274].

64. While in “user interface reconfiguration mode,” “an object removal badge (e.g., 5010 in FIG. 5B)” appears and “when the device detects activation of the object removal badge (e.g., tap gesture 5011 in FIG. 5B).” EX1004, [0218]. “In some embodiments, instead of, or in addition to, the selectable user interface object removal region described above, the device displays an object modification target associated with one or more of the selectable user interface objects.” EX1004, [0273]. This object modification target is displayed while in “user interface reconfiguration mode.” EX1004, [0274]. While in this mode the device may “detect[] a request to activate the [orange] object modification target [5194] for a respective selectable user interface object (e.g., tap gesture 5196 at a location on the touch screen 112 that corresponds to a location of an object modification region for

action icon 5002-37), the device displays an object modification dialogue. For example in FIG. 5LLL the device displays a [purple] pop-up dialogue 5198 that provides the user with options for modifying the selectable user interface object.” EX1004, [0274]. And then a user is able to delete the “action icon 5002-37” by a “tap gesture 5200 on a [red] delete button.” EX1004, [0274].



EX1004, Figures 5KKK-LLL (annotated).

B. U.S. Patent Application Publication No. 2004/0051726 to Martyn (“Martyn”)

65. Martyn describes a non-touch screen device having six equally valid calling methods for a “‘snap-shot’ view.” Martyn does not express a preference for one calling method over another calling method. Martyn identifies an “*innovative*

summary window functionality.” EX1005, [0017]. Martyn’s “‘snap-shot’ view ... brings together, in one summary window, a limited list of common functions and commonly accessed stored data which itself can be reached directly from the main menu listing some or all applications.” EX1005, [0010].

66. Martyn uses several different terms interchangeably for example “App Snapshot,” “summary window,” and “snapshot view” are all referring to the same aspect of Martyn. EX1005, [0009-0010], [0015], [0017-0026], [0028-0031].

67. The first method Martyn describes is a “timeout” calling method and uses a *two-step* process. First, navigating a highlight to a desired application and, second, “call[ing] or fir[ing]” the Snapshot. EX1005, [0017]. A highlight indicates to a user what application subsequent selection actions would correspond to. For example, a highlight on the messages app would indicate that the message app is selected. A user starts by looking at a device in a “main view,” such as an “Application Launcher view [that] present[s] [] a standard scrolling list of application[s].” EX1005, [0016]. Seeing multiple application at one time enables the user to effectively have a visual cue for what application is currently selected and what application would be next selected depending on a navigation choice. A “user may take conventional navigation steps, such as scrolling, to move the highlight.” EX1005, [0016]. Then “should the highlight rest on the name of an application in

the App Launcher for a certain amount of time” the Snapshot is “called” or “fired.” EX1005, [0017].

68. Martyn also describes that two “[o]ther selection processes ... (e.g. voice activation, softkey selection etc).” EX1005, [0017]. Martyn recognizes that these are selection processes for triggering its Snapshot.

69. Martyn also gives three other “example[s].” EX1005, [0017]. These other examples include “using a right scroll function (if the mobile telephone has 4-way Scrolling capacity); a press and hold of the Select function when an application name is highlighted; or using a right cursor key on a highlighted application name.” EX1005, [0017]. Since Martyn’s device does not describe a touch-screen device, the selection function would correspond to a physical input method for the device. For example, a button or the depression of a scroll wheel.

70. Of these, Martyn describes a highlight explicitly or implicitly for all but the voice activation method. A softkey selection requires the pressing of a button. As described in Martyn there would need to be some indication of what application is being selected for the Snapshot when pressing the softkey. Accordingly, a highlight navigation is still required for the soft key selection. Although the function of the soft key may have additional functionality, when used for calling Martyn’s Snapshot there would need to be consistent situations or contexts in which the

softkey is associated with the launch of the Snapshot and as described by Martyn that would be a highlight indicating a desired application.

71. Consistent with the process outlined above for a timeout, “a press and hold of the Select function” does not stand in isolation rather, it is “a press and hold of the Select function *when an application name is highlighted.*” Martyn describes a user may press and hold a selection function to launch Martyn’s Snapshot only after navigating a highlight to the desired application. Without a highlight the user is unaware of what application is being called by a press and hold of a select button.

72. Martyn diagrams the first calling method in Figures 1 and 2, as annotated below with a red highlight and a blue summary window.



Figure 1

Navigated highlight
rests and the
Snapshot is
launched

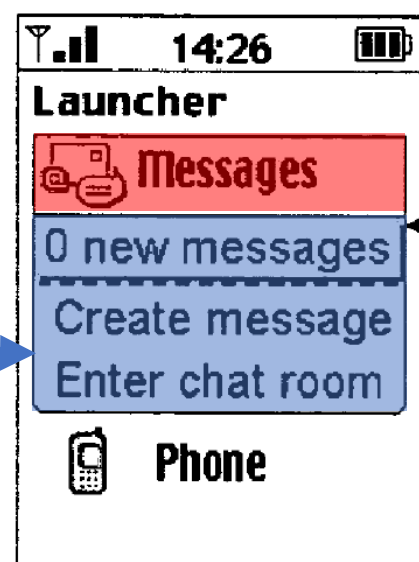


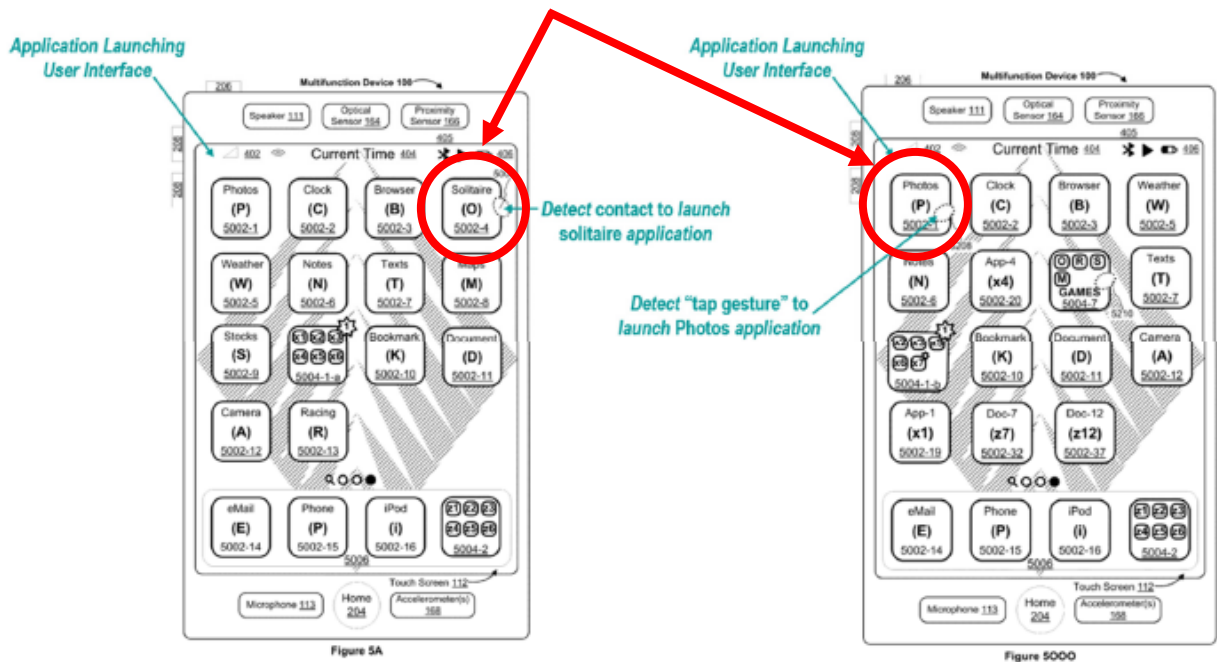
Figure 2

X. Petitioner fails to properly map “the first application icon.”**A. Petitioner Fails to Consistently Map “The First Application Icon”**

73. Limitations [1.g], [63.c], [124.b] recite “detect[ing] a first single-finger touch input” “at a location on the touch-sensitive surface that corresponds to a first application icon.” Limitations [1.l], [63.h], [124.g] then recite “moving the first application icon,” which must, under the plain meaning, be the “the first application icon” mapped for Limitations [1.g], [63.c], [124.b].

74. Petitioner provides two alternative mappings of Limitations [1.g], [63.c], [124.b], mapping the “first application icon” to either: (a) the Solitaire icon or (b) photos application icon. Pet., 26-27, 61. Petitioner does so first through prose, parentheticals, and emphasis: (a) “Chaudhri explains that ‘the device *detects* a contact 5008 **with the solitaire application icon**’ and performs a responsive action (e.g., *launching the application* or entering the reconfiguration mode) based on the characteristics of the touch;” (b) “as shown in Figure 5000, ‘*detecting* a tap gesture 5208 at a location on the touch-sensitive surface that corresponds to the **photos action icon** 5002-1’ causes the device to ‘*launch[] the photos application*.’” *Id.*, 26 (bold italics emphasis in original, highlighting added). Petitioner affirms these two mappings through arrows and text annotating Figures 5A and 5000:

Petitioner's two mappings of the "first application icon" for Limitation [1.g]

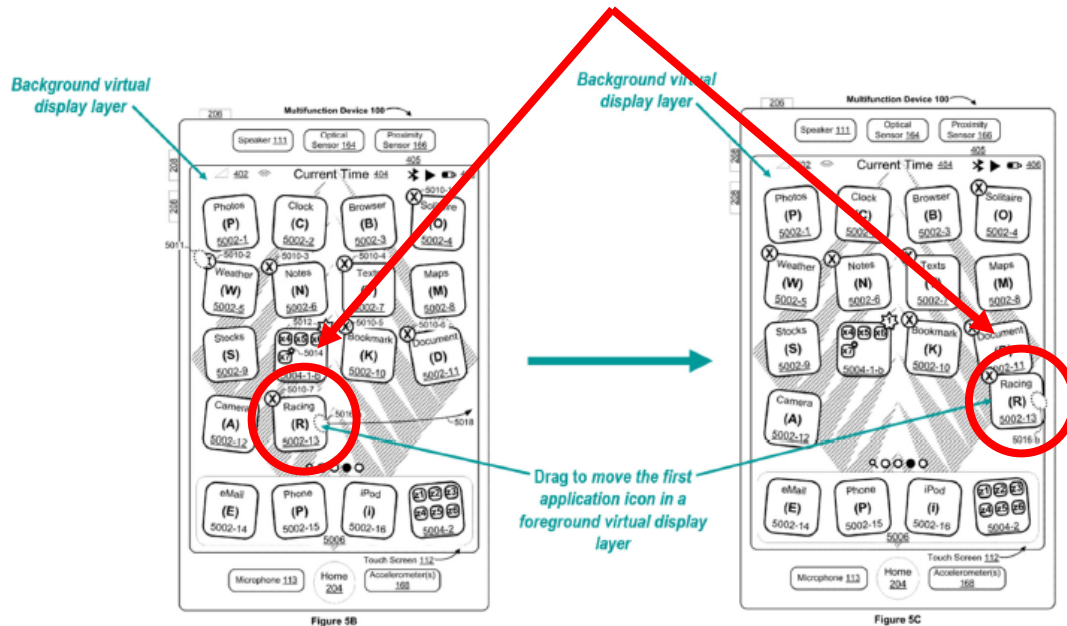


EX1004, Figures 5A (annotated), 5000 (annotated); EX1003, ¶85.

Pet., 26.

75. For Limitations [1.l], [63.h], [124.g], Petitioner does not identify “moving” either the Solitaire or Photos icon. Pet., 36-38, 61. Instead, Petitioner states that the device “detects a touch ‘at a location that corresponds to a “racing” [application] icon’ coupled with ‘subsequent movement’ of the touch ‘to the edge of the touch-sensitive surface,’ the *application icon* is likewise ‘*moved* to the edge of the display.” *Id.* (bold italics emphasis in original, highlighting added). Petitioner affirms this mapping through arrows and text annotating Figures 5B-5C:

Petitioner's mapping of the "first application icon" for Limitation [1.i]



EX1004, Figures 5B-C (annotated); EX1003, ¶100.

Pet., 36.

76. Having previously established the "first application icon" as either the Photos icon or the Solitaire icon, Petitioner's mapping to Chaudhri's disclosure of moving the Racing icon is insufficient to meet the plain language of the Limitations [1.i], [63.h], [124.g] of moving "the first application icon." Petitioner provides no alternative mappings that could meet the plain language. Pet., 36-38. Thus, Petitioner fails to map Limitations [1.i], [63.h], [124.g].

B. Petitioner Fails to Identify Moving “The First Application Icon” Based on “The First Single-Finger Touch Input”

1. Petitioner’s Mapping to Determining A Different Touch Input After “The First Single-Finger Touch Input Concludes” Meets Movement Criteria Does Not Meet the Claim Language

77. As discussed above in ¶¶50-54, above, the plain language of Limitations [1.l], [63.h], [124.g] “a determination that **the** first single-finger touch input meets the one or more movement criteria” requires determining a portion of the touch that began with the initial touch detected in Limitations [1.g], [63.c], [124.b] meets the movement criteria.

78. Similarly, because “**the** first single-finger touch input” concludes upon lift off, determining that a different touch occurring after “**the** first single-finger touch input” concludes meets the movement criteria is insufficient to meet the plain claim language. As discussed above, the claims describe logic where liftoff is required for the duration to cease so that the length of the duration may be appropriately evaluated against a time threshold.

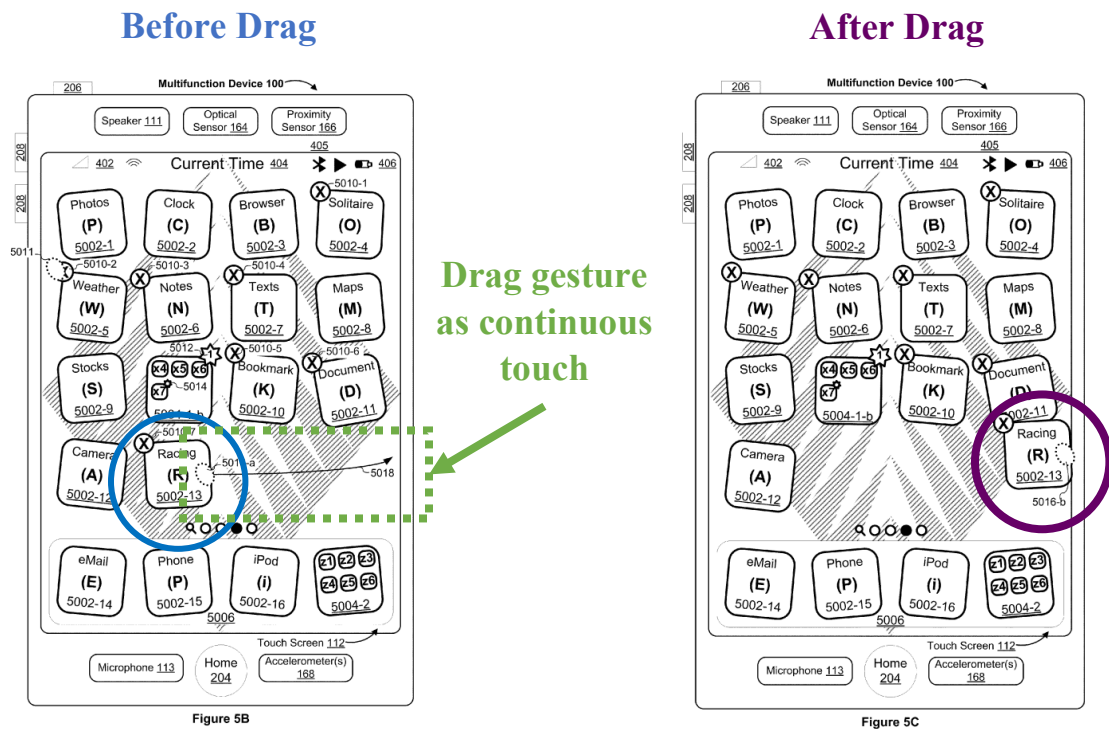
79. Petitioner does not contend that its primary mapping meets the claim requirement. Pet., 35-38. Nor can it. Chaudhri’s discloses “the device detect[ing] a *contact 5008* with the solitaire application icon 5002-4 for more than a predetermined period of time (e.g., 2 seconds) and, in response, the device enters a user interface reconfiguration mode.” EX1004, [0217]; Pet., 33-34 (citing same for Limitation [1.k]). Then, separately, *after* the device is in reconfiguration mode, the

device detects a *separate* “*contact 5016-a* on the touch-sensitive surface (e.g., touch screen 112) at a location that corresponds to a “racing” action icon 5002-13, and subsequent movement 5018 of the contact to the edge of the touch-sensitive surface.” EX1004, [0220]; Pet., 36 (citing same for Limitation [1.1]).

80. A POSITA would understand that the reference to two distinct contacts and separately detecting those contacts refers to two separate discontinuous touches. Only by referencing and describing the *same contact* would a POSITA understand that contact to be continuous as discussed above in ¶¶50-54.

81. Furthermore, in contrast to the two separate contacts that the petitioner maps. Chaudhri gives an example of a continuous contact with a touch as a dotted circle and an arrow originating out of that circle in the context of a drag gesture. EX1004, [0220], Figures, 5B-C. “The dragging, for example, comprises a touch (or contact) on the displayed object for a predetermined phase, a movement of the touch across touch-sensitive display 112, and lift-off of the touch (touch end).” EX1004, [0164], *see also* [0101]. This drag gesture is shown in Figures 5B and 5C, reproduced below. “In FIG. 5B, the device detects a request to move a” “user interface object to an edge of the screen.” EX1004, [0220]. The request is initiated by “a contact 5016-a on the touch-sensitive surface” at the location of the ““racing’ action icon 5002-13” where a “[green] subsequent movement 5018 of the contact to the edge of the touch-sensitive surface” relocates the action icon 5002-13 as shown in Figure 5C.

EX1004,



EX1004, Figures 5B-C (annotated).

82. Accordingly, Petitioner's mapping to Chaudhri's disclosure does not meet Limitations [1.l], [63.h], [124.g].

2. Petitioner Proposes Further Modifying Without a Real Motivation

83. Aware that its primary mapping does not meet the plain language requirements of Limitations [1.l], [63.h], [124.g], Petitioner proposes two alternative obviousness modifications that contradict with each other and are both founded upon a mischaracterization of Chaudhri to create a reason to seek modification that does not exist. Pet., 38-39, 61-67. Petitioner asserts that "[a] POSITA would have

recognized that there are only three apparent design options *to implement Chaudhri's teachings* for using a drag gesture to move an application after shifting from the 'normal mode' to the 'reconfiguration mode' via a touch-and-hold gesture." Pet., 38.¹ But the premise of Petitioner's arguments that it is free to select "design options to implement Chaudhri's teachings" assumes Chaudhri is silent and thus there is a "design choice" to fill in. But as discussed directly above in ¶¶77-82, that premise is not correct because Chaudhri explicitly describes two non-continuous touches for the touch Petitioner maps to earlier claim elements and the touch Petitioner maps to meeting Limitations [1.l], [63.h], [124.g].

84. For example, Petitioner's mapping for Limitations [1.l], [63.h], [124.g] introduce a racing application icon that is not previously shown to have been touched for any of the earlier Limitation mapping. In fact, a different application icon is shown to be mapped (namely a solitaire application icon and photos application

¹ "First, the device and interface could be designed so that the user must keep their finger down ... Second, the device and interface could be designed so that the user must break contact ... Third, the device and interface could be designed so that the user may input the two gestures either as a single continuous touch or as consecutive touches." Pet., 38-39.

icon) as described above in ¶¶ 73-74. Without touching the same application, the touch cannot be continuous.

85. Petitioner provides no reason beyond the inaccurate assertion that Chaudhri is silent for why a POSITA would even begin to explore modifications. Pet., 38-39, 61-67. Accordingly, both Petitioner's Ground 1 and 2 modifications fail because Petitioner identifies no motivation to modify Chaudhri's disclosure. With the understanding of Chaudhri forwarded by Petitioner there is simply no reason to suggest a design implementations that ignore the content of the reference provided by Petitioner.

86. Petitioner's Ground 1 and 2 mappings each also fail independently. Petitioner's assertion that "the most natural way to implement Chaudhri's teachings would be to select the third" option "because the device and interface would [be] easier to use" (Pet., 39) inaccurately assumes Chaudhri's silence and provides no analysis of Chaudhri's actual implementation and why that is not the proper selection for Chaudhri. Petitioner has not provided a reason for departing from Chaudhri without accurately describing Chaudhri in the first instance.

87. Turning to Ground 2, according to Petitioner, "Os teaches a 'continuous multipart gesture' that includes a 'tap and hold gesture' combined with 'subsequent lateral movement of the contact,' all while the user's finger remains down on the touch screen." Pet., 66. In the terminology Petitioner coined in Ground 1,

Petitioner's characterization of Os matches the "first" option ("the user must keep their finger down"). *Id.*, 38-39. Given Petitioner already asserted that that a POSITA would select the "third" option (*id.*, 39), any argument that a POSITA would be motivated to select the "first" option rings hollow.

88. Further, a required continuous touch cannot be reconciled with how Petitioner characterizes Chaudhri. A touch cannot both be continuous and discontinuous. Additionally, logic for detecting a continuous touch does not detect a discontinuous touch in the same manner.

89. As mapped by Petitioner, Chaudhri explicitly shows entering reconfiguration mode by touching one application icon, the solitaire icon, and then performing a second touch and drag to move a second icon, the racing icon. Requiring the same continuous touch for both entering reconfiguration mode via touch of one icon and moving an icon would remove the functionality shown across Figures 5A-5C that Chaudhri explicitly discloses. A POSITA would not remove a functionality already described by Chaudhri that allows for user flexibility, let alone doing so without a motivation. As mapped by Petitioner, Chaudhri discloses allowing the user to enter reconfiguration mode with a touch of one application icon and then lifting a finger to touch and move a second application icon.

XI. A POSITA would not have taken Petitioner’s desired calling method for calling Martyn’s Snapshot

A. Petitioner Contorts the References to Reach the Purported Combination

90. Petitioner’s proposed combination begins with Chaudhri’s device that already: (1) “transitions from the ‘normal operation mode’ to a ‘reconfiguration mode’ *when the user executes a touch-and-hold gesture” on an application icon* (Pet., 21), and (2) includes functionality, in reconfiguration mode, for “displaying a contextual .. sub-menu ... with options” and “presenting commonly used application functions” (*id.*, 18-19) *when the user taps a circle placed in the corner of the application icon* (EX1004, [0273]-[0274]). Ignoring these existing functions, and with the unapologetic goal “to successfully arrive at the ’758 patent’s claimed invention” (Pet., 16), Petitioner proposes a series of modifications inspired by nothing but hindsight.

91. First, Petitioner proposes “integrat[ing] Martyn’s teachings about” (1) “providing a contextual shortcut sub-menu in the form of a ‘summary window’ common application functions (an ‘App Snapshot’)” and (2) calling the App Snapshot through one of a series of methods Martyn discloses: “a press and hold of the select function *when an application name is highlighted.*” Pet., 16-17; EX1005, [0016]-[0017]. But, as discussed below, Petitioner identifies motivations only for the former, not the latter. Second, Petitioner does not propose actually using the

selected gesture from Martyn, but rather using a different gesture that the Petition alleges in a single conclusory sentence to be “the analog to Martyn’s press-and-hold input on a touch screen.” Pet., 16.

92. Third, Petitioner proposes further modifications because “the POSITA would have sought a solution to the problem of distinguishing between the different gesture commands”—*the very problem created by the first modification*. *Id.*, 21. To resolve this conflict, Petitioner proposes assigning one gesture a long touch and another gesture a longer touch. *Id.*, 21-22. And, fourth, Petitioner proposes, not uncoincidentally to match the claim, calling the App Snapshot with a long touch, thereby *countermanning* Chaudhri’s disclosure of entering reconfiguration mode through a normal long touch on an application icon and, instead, requiring the user first call up the App Snapshot and then continue with a *longer* touch. *Id.* In support of those third and fourth modifications, Petitioner nominally points to Griffin, but does not identify any disclosure in Griffin of any motivation or advantages—just that Griffin happens to mention time thresholds. *Id.*

93. As discussed below, Petitioner’s series of modifications ignore the existing disclosures of Chaudhri that teach away from assigning a long touch to calling the App Snapshot and already disclose a gesture for calling up what Petitioner describes as a context menu. Moreover, actually applying Petitioner’s selected call up method from Martyn or applying numerous other alternatives Martyn discloses

results in modifications not taught away from (and that do not meet the claims). And, finally, even if a POSITA assigned a long touch to calling up the App Snapshot, Chaudhri discloses alternative methods for entering reconfiguration mode that would also avoid a conflict.

94. Accordingly, taking into account Chaudhri and Martyn's actual disclosures, a POSITA would be motivated to implement Martyn's App Snapshot using numerous other call up methods, none of which match the claims.

B. A POSITA would not have been motivated to choose a specific method of calling Martyn's SnapShot

95. Petitioner attempts a sleight of hand, relying on Martyn not just for the App Snapshot, but also for its disclosure of how to call the App Snapshot, but presenting motivations that address only the former. Pet., 15-19. Specifically, each of the four "motivations" Petitioner provides are limited to why a POSITA would include Martyn's App Snapshot. *Id.*, 16-19. Petitioner provides no reasons a POSITA would look to Martyn for its disclosure of how a user would call up the App Snapshot in Chaudhri, let alone why a POSITA would choose the specific option Petitioner selects over others Martyn discloses and that are otherwise available. *Id.*

96. Petitioner's first three motivations to combine Chaudhri and Martyn exclusively discuss the benefits of a summary window itself, like Martyn's Snapshot, and not a specific calling method for that summary window.

97. Petitioner’s first motivation is that “it would have been axiomatic for a POSITA” to “employ[] a contextual shortcut sub-menu to augment a ‘main menu listing some or all applications’ of a mobile device” “in the context of Chaudhri’s user interface” because “sub-menus were exceedingly well-known solutions to [an] equally well-known problem.” Pet., 16-17.

98. All of Dr. Cockburn support for this first motivation—EX1003, ¶¶43-48, 64-65—could only establish that “*providing a sub-menu* of selectable options corresponding to commonly used application functions *was a ubiquitous feature* in computer GUIs by 2011,” not that a specific method for calling those sub-menus was ubiquitous. Pet., 16.

99. Dr. Cockburn explains that “menus were a common feature” because of the problem they solved. EX1003, ¶¶43-44. Dr. Cockburn also explains that “*menus* are an effective, time-tested solution to the problem in user-interface design of presenting available actions and functionalities for quick and efficient selection by the user” because “‘users no longer need to refer to manuals or memorize available options’ [and] ‘instead, the choice can be called up at a moment’s notice whenever needed.’” EX1003, ¶44. This problem-solution pair only addresses *what* a user sees and not *how* a user sees it. Menus provided visual cues to a user to have a options to select from. Memorization of all possible applications on a device defeats the purpose of a device that can visualize those applications and options.

100. Dr. Cockburn cannot motivate the commonality or longstanding use of Petitioner’s desired calling method when the examples that Dr. Cockburn points to describe a variety of different calling methods in addition to a touch-and-hold. Without a strong consistent signifying feature, it cannot be said that there is one calling method for menus or context menus. A user interface needs flexibility to be suited to implement features in an effective way. After all, the touch screen itself is a way of providing that flexibility. Considering narrowly that a specific calling method must be used ignores the advances of devices.

101. For example, Dr. Cockburns points to “‘list picker’ menus,” “text editing menus,” a “contextual menu for Maps application,” and a “contextual menu for Voice Memo application.” EX1003, ¶45. All of which were respectively accessed by taps. EX1022, 136 (“A user makes a list picker pop up by tapping the button or label that controls it.”); EX1045, 34 (“Tap the insertion point to display the selection buttons.”); EX1045, 117 (“Tap” a map page reveal icon); EX1045, 128 (“Tap Trim Memo.”). Dr. Cockburn further admits “GUI menus” were implemented “in in a variety of different forms (e.g., pop-up, drop down, toolbar, etc.)” without mention a single calling method. EX1003, ¶64. Each of these demonstrate different implementations of calling methods, not one longstanding calling method.

102. Petitioner’s second motivation is that “providing a sub-menu of options for invoking common application functions, as Martyn teaches, would allow the user to accomplish recurring tasks more quickly and efficiently (e.g., with fewer user inputs).” Pet., 17. The Petition calls this “usability” and defines it “in this context [as] the degree to which a given user interface efficiently facilitates the user’s desired task.” Pet., 17. The Petition then states that this increased “usability” was “a key differentiator amongst competing mobile devices” and that it “would be more attractive to consumers.” Pet., 17.

103. Much like the Petitioner’s first motivation, the petition glosses over the difference between a *what* (the menu itself) and *how* (calling the menu). Petitioner’s definition and expert support for usability all focus on how a menu simplifies and reduces user input by bringing together “options for invoking common application functions.” Pet., 17. Dr. Cockburn explains that “[r]educing the user’s cognitive load by displaying common functionalities in a sub-menu would make Chaudhri’s device and its user interface for a ‘menu of applications’ easier to use.” EX1003, ¶66. “[T]he user’s memory load” is “minimize[d] by making objects, actions, and options *visible*” and “provid[ing] shortcuts to the most frequent tasks,” not by any specific way of accessing the menu that brings together those options. EX1039, 3 (emphasis added); EX1003, ¶66. As described above in ¶99, menus and visible applications

prevent a user from having to memorize everything on their device or how to reach desired functions for those applications.

104. The explanation provided by the Petition is limited to the visibility and collectiveness offered by menus and does not provide a motivation to select Petitioner’s desired calling method—among the several options listed in Martyn. A calling method does not group applications together, that is the function of the menu. Likewise two calling methods can equally make a menu visible.

105. Petitioner’s third motivation is that “Martyn identifies specific ‘advantages’ of the ‘App Snapshot’” and given “the similarities between Chaudhri and Martyn’s respective multi-functional mobile devices, the POSITA would have sought to predictably improve Chaudhri’s interface for accessing application functions in the same way Martyn did—i.e., via a contextual shortcut sub-menu.” Pet., 17-18.

106. Petitioner’s third motivation largely repeats the same argument as Petitioner’s first two but by pointing to disclosures in Martyn. Pet., 17-18. As the Petition summarizes “[w]here other techniques require the user to navigate between ‘many layers’ and/or only show ‘one option...at any one time,’ Martyn’s App Snapshot ‘brings together, in one summary window, a limited list of common functions,’ which ‘yields many advantages in ease and speed of navigation.’” Pet., 18. Again, Petitioner merely points out that the “summary window” is advantageous

because it brings together in one location multiple options that were previously spread out across the device. Pet., 18.

107. Additionally, Martyn does not discuss the speed of navigation in relation to *how* the “summary windows” is called. EX1005, [0015]. Martyn simply does not discuss a preferable calling method of one over another. Undoubtedly, some users may be able to click faster than a timeout, yet Martyn offers both as achieving ease of speed and navigation. This is because it is the menu described in Martyn that offers those advantages not the calling method.

108. The utility of a context menu is that it is an alternative from having to navigate through many layers of an application to reach a desired function. Layers refer how a user must navigate through many different sets of options to reach a desired result. For example, launching an email application and then having to click reply or forward. The context menu brings both of those options to the forefront without requiring the launching of the email application.

109. Of the available calling methods from Martyn, the “press and hold of the select function when an application name is highlighted” method requires an application name to be highlighted, otherwise a user does not know which application that Martyn’s Snapshot will be called for. This is the a two-step process as described earlier, in ¶67, in relation to the timeout calling method. If minimizing

time is desired then a POSITA would have selected a calling method described in Martyn that does not contain this step.

110. Petitioner’s fourth motivation is that a “POSITA would have been prompted to pursue a combination with Martyn based on Chaudhri’s suggestions to do so.” Pet., 18. The Petition explains that Chaudhri’s suggestion is “the *notion of displaying a contextual* ‘pop-up dialogue’ *sub-menu* that ‘provides the user *with options* for modifying the selectable user interface object’” and that suggestion “would have prompted a POSITA to seek out and apply the teachings from other references in the same field, like Martyn, that describe analogous implementations of contextual sub-menus to augment a user interface for a multi-functional mobile device.” Pet., 18 (emphasis added). The Petition further states that “the straightforward discussion in Martyn about *presenting* commonly used application functions as selectable sub-menu options in an App Snapshot/summary window is a natural and obvious extension of Chaudhri.” Pet., 18-19 (emphasis added).

111. Again Petitioner’s motivation is limited to the context menu itself and not a calling method. Petitioner characterizes Chaudhri’s suggestion as related to “displaying a contextual .. sub-menu ... with options” and “presenting commonly used application functions.” Pet., 18-9; EX1003, ¶69. Like Petitioner’s other motivations, these statements describe the advantages of a contextual menu—i.e. bringing together many applications and functions that a user desires. As described

above in ¶¶86-99. These statements, again, do not opine on *how* to call a contextual menu.

112. Additionally, Dr. Cockburn concedes that the other paragraph cited in the Petition and expert declaration likewise do not provide a motivation to use Petitioner’s desired calling method. EX2007, 59:15-17 (“A. [EX1004, [0317]] itself contains no explicit mention of how the contextual menu is -- is posted.”). This paragraph merely mentions context menu in passing and does not describe a preferred calling method for said context menu. Chaudhri states “[a]s another example, the device could receive a first input selecting a first item, a second input selecting a second item and a third input corresponding to a command to perform a new folder creation operation (e.g., a click on a “new folder” command in a file browser or a contextual menu, etc.).” EX1004, [0317]. This statement merely describes inputs generically for a context menu not any specific calling method.

C. Chaudhri Already Discloses A Calling Method for A Context Menu

113. Petitioner contends that Chaudhri discloses “displaying a contextual .. sub-menu ... with options” in reconfiguration mode and “presenting commonly used application functions.” Pet., 18-19 (citing EX1004, [0247]); EX1003, ¶69. Specifically, Chaudhri discloses “in FIG. 5KKK each of the selectable user interface objects has an object modification target associated with it (e.g., action icon 5002-37 has a respective object modification target 5194).” EX1004, [0246]. A user calls

the context menu, “object modification” “pop-up dialogue 5198,” through a green “tap gesture 5196 at a location on the touch screen” corresponding to a circle placed in the corner of the application icon corresponding to “object modification target 5194.” EX1004, [0246]-[0247]; EX2007, 54:17-55:7 (“So the tap gesture is one example of the mechanisms used to activate the object modification target”).

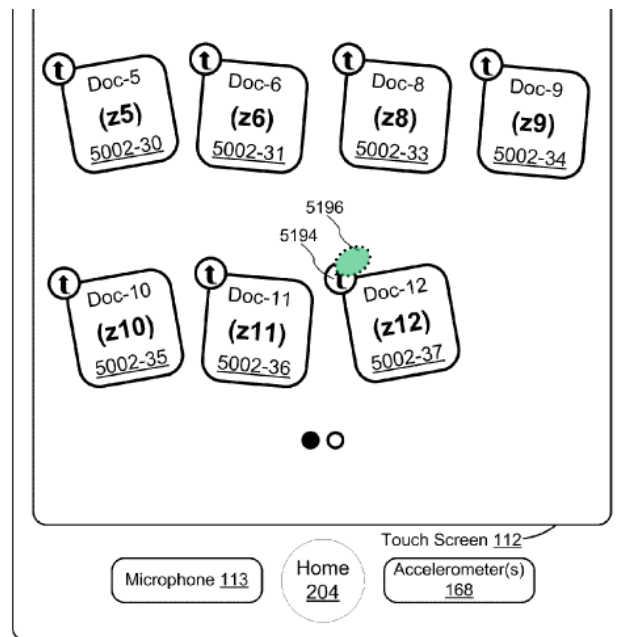


Figure 5KKK

EX1004, FIG. 5KKK (annotated).

114. A POSITA would understand this technique is readily applicable to calling an App Snapshot in normal mode. As Chaudhri describes, a user calls an “object modification” “pop-up dialogue 5198,” (i.e. a context menu) by tapping a region associated with a desired application icon. EX1004, [0246]-[0247]. A method for calling a menu in a particular context is equally applicable whether in reconfiguration mode or normal mode. A POSITA would appreciate that this object

modification target is desirable not because it is limited to the reconfiguration mode but rather because it permits a user to reach contextual actions or information associated with an application. A feature equally desirable in normal mode. In other words, the object modification target is desirable, as indicated by Chaudhri, method to reach a context menu.

115. As annotated on Figure 5A, below, a POSITA would have understood that target regions (shown as red circles circle with “s” for “snapshot”) could be included in the upper lefthand corner on icons and be associated with an App Snapshot when in normal mode just as the “object modification target” regions shown in Figure 5KKK. EX1004, [0273]-[0274]. Describing an object modification target in functional terms conveys to a POSITA the ability to use it to perform different functions and because the function it provides gives context to the target a POSITA would understand that additional context could be added for the target and reached in the same way.

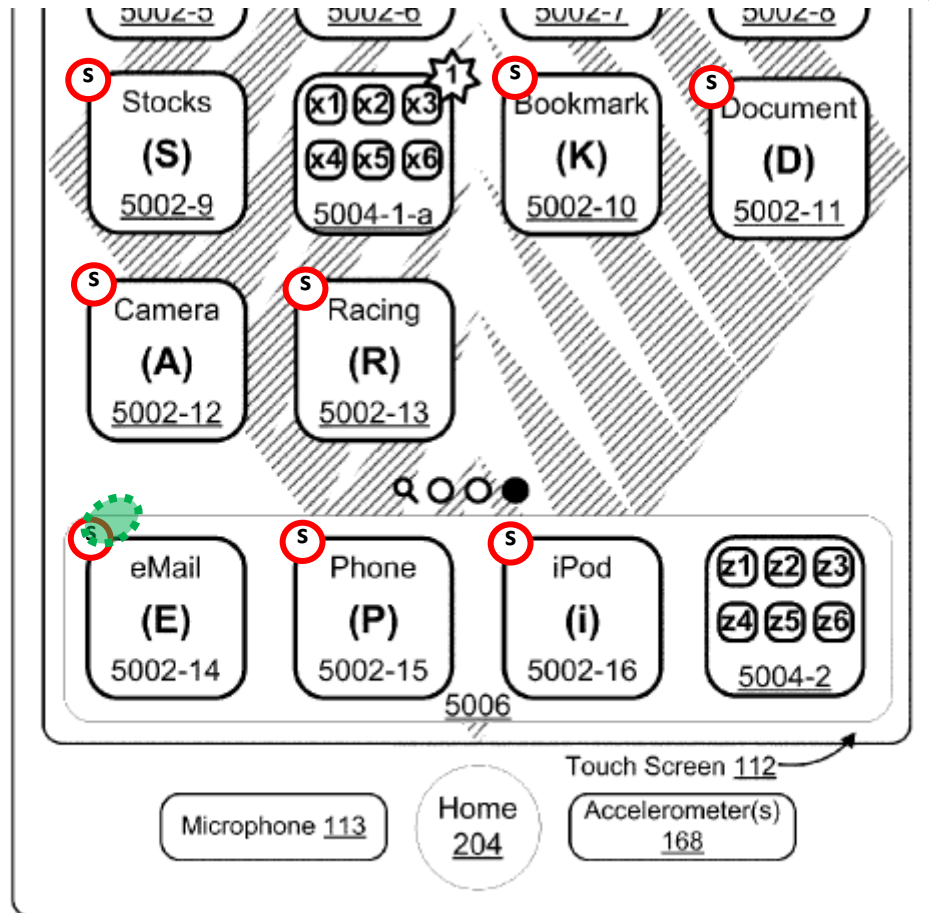


Figure 5A

EX1004, FIG. 5A (annotated).

116. Because the App Snapshot is called from normal mode, the “object modification target” regions shown in Figure 5KKK, in reconfiguration mode, would not cause any conflict. A user may call Martyn’s Snapshot menu via a tap in normal mode. This does not overlap with a touch-and-hold of an application icon to enter reconfiguration mode. First, the duration of the contacts are different. Second, the locations of the contacts are different. The touch-and-hold is at the application

icon, while the tap is not on the application icon but rather at a specific object modification target.

117. As further annotated on Figure 5A, below, Martyn's App Snapshot shown in purple would appear when a user tapped the target region defined above as a red circled s.

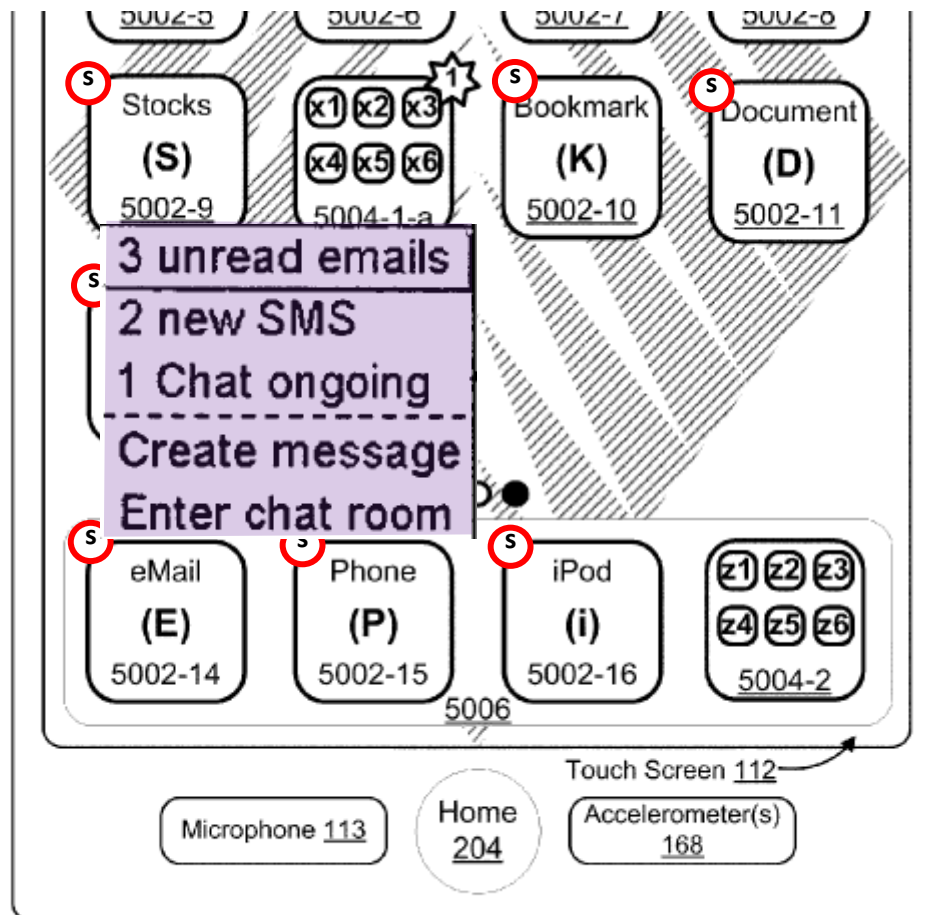


Figure 5A

EX1004, FIG. 5A (annotated after tap gesture).

118. Chaudhri's disclosure of a readily applicable method to call up a context menu teaches away from turning to Martyn's disclosure for that functionality

because there is no need to do so. Chaudhri already express a desirable method for calling a context menu.

119. Further, the method Chaudhri already discloses is superior to Martyn's calling methods in multiple ways. Indeed as Dr. Cockburn stated in his declaration: "Martyn's contextual sub-menu 'summary window' is related to selecting and launching the application (i.e., assisting the user with selecting a desired application functionality), the POSITA would have appreciated that this operation should likewise be part of the 'normal' mode." EX1003, ¶¶77. In this way, a user would receive the most value out of accessing the context menu directly in the normal mode. Accordingly, a POSITA would appreciate the most direct method for doing so would be the most desirable.

120. First, this method, does not present any conflict with an already assigned method in Chaudhri.

121. Second, a tap on a target region is faster and simpler than the two-part (first navigate a highlight to an application and then long press a select function button) calling method Petitioner proposes selecting from Martyn or the long touch Petitioner asserts is the "analog" to Martyn's actual disclosure. A tap on a target region does not involve any navigation nor durational touch exceeding a first time threshold. This reduces both the number of steps and time required to reach Martyn's Snapshot.

122. The advantages realized by the calling method described in Chaudhri also teaches away from taking a specified calling method from Martyn.

D. The Long Touch Gesture Is already Assigned to a Different Gesture Teaches Away From Assigning That Gesture To Calling The App Snapshot

123. Petitioner readily admits that its proposal to assign a touch and hold gesture on an application icon to calling an App Snapshot that Chaudhri had already assigned to a different function creates a “*problem* of distinguishing between different gesture commands.” Pet., 21.

124. A POSITA would not be motivated to select an implementation that *creates a problem* when doing so does not solve any unsolved problem or have any benefit over other solutions that do not *create a problem*. A POSITA would not create additional problems without exhausting potential solutions because creating an additional problem can create an endless problem solution cycle.

125. Rather, the fact that Chaudhri already assigned the touch and hold gesture on an application icon to a different function *teaches away* from assigning that gesture to a second function. An implementation that *creates a problem* is not a suitable option.

E. Numerous Available User Gestures That Are Consistent, Rather Than Conflict With Chaudhri and Martyn’s Teachings, Further Teach Away From Selecting a Gesture That Creates A “Conflict”

126. As discussed above in ¶¶113-122, Chaudhri itself already discloses a user gesture—a tap on a target region in the top corner of an application icon—to call a context menu that does not conflict with Chaudhri’s other teachings. Moreover, as discussed below, applying the disclosures of Martyn and Chaudhri leads to numerous modifications that do not create a conflict. The availability of all of these options that do not create a conflict further teaches away from selecting a user gesture that creates a conflict. A POSITA would be discouraged from seeking an option that would create additional problems when presented with other solutions that do not create conflicts to be resolved.

1. Consider Martyn’s “press and hold of the select function when an application name is highlighted” as disclosed avoids conflict

127. As discussed in paragraphs ¶¶65-72, 90-94, above, Martyn does not disclose calling the App Snapshot based on a touch and hold on an application icon. Rather, Petitioner points to Martyn’s disclosure of “a press and hold of the select function when an application name is highlighted.” Pet., 16-17; EX1005, [0016]-[0017]. Petitioner then, in a single conclusory sentence asserts the separate gesture is “the analog to Martyn’s press-and-hold input on a touch screen like Chaudhri’s.” Pet., 16. The cited testimony of Dr. Cockburn is nothing more than a repeat of the conclusory claim in a footnote. EX1003, ¶63, n.3.

128. Contrary to Petitioner’s conclusory assertion, applying Martyn’s “press and hold of the select function when an application name is highlighted” to Chaudhri does not result in a touch and hold on an application icon. Martyn describes launching its Snapshot with this calling method in a *two-step* process. *Supra* §IX.B; EX1005, [0016]. First, a user navigates to a desired application using “[s]tandard highlight functions” before calling the App Snapshot. The method Petitioner relies upon and explicitly states “a press and hold of the select function *when an application name is highlighted*,” indicating the required first step of navigating to the application. EX1005, [0016-0017]. EX2007, 39:3-12.

129. For the press and hold of a select function to have meaning to a user, there must be some indication to the user which application will be triggered when pressing and holding a select function. The highlight provides necessary information to the user. Without a visual highlight, a user does not have an indication of what will happen when pressing and holding a select function.

130. In applying Martyn’s disclosure to Chaudhri, a POSITA would first address the navigating step. The above described information conveyed by a highlight is enabled by a change in that highlight. If a highlight was not able to be associated with different applications (i.e. navigating), the highlight would not convey the desired information to the user for knowing what will happen when a user presses and holds the select function.

131. Chaudhri discloses “[i]n some embodiments, device 100 may include a physical or virtual wheel (e.g. a click wheel) as input control device 116.” EX1004, [0093]. Chaudhri further disclose that “[a] user may navigate among and interact with one or more graphical objects (e.g. icons) displayed in touch screen 112 by rotating the click wheel or by moving a point of contact with the click wheel.” EX1004, [0093]. Applying Martyn’s disclosure, a POSITA would implement the initial step of navigating the highlight using Chaudhri’s click wheel. Chaudhri’s click wheel provides the ability to navigate between applications in a way that conveys to a user the necessary information for launching Martyn’s Snapshot. A click wheel allows a user to change which application is selected and consequently which application Martyn’s Snapshot will be launched for.

132. Next turning to the function key, Chaudhri further discloses using the click wheel for this purpose: “The click wheel may also be used to select one or more of the displayed icons. For example, the user may press down on at least a portion of the click wheel or an associated button.” EX1004, [0093]. Thus, to implement “a press and hold of the select function when an application name is highlighted” in Chaudhri, a user would use the click wheel to navigate a highlight to the desired application icon and then “press down on at least a portion of the click wheel or an associated button” in a long press to call up the App Snapshot. EX1004, [0093].

133. This process follows the same two-part process described in Martyn. The click wheel performs standard highlight functionality and further offers select key functionality either within the click wheel or with an associated button.

134. Given Chaudhri specifically discloses navigation and select key functionality, a POSITA would use that functionality to implement Martyn's disclosure. Chaudhri's disclosure of a click wheel provides both of the elements described in Martyn: a navigation and a selection functionality. Petitioner's conclusory claim that a different gesture is an "analog" ignores Chaudhri's disclosure, specifically the availability of this functionality. A POSITA has no reason to resort to an analogy when the references align in their disclosures.

135. Further, this actual application of Martyn's teaching constitutes another example of method of calling an App Snapshot that does not create a conflict with Chaudhri's existing disclosures. A physical or virtual click wheel do not involve a duration based input at an application icon. Therefore, no conflict would be created between the click wheel calling methods and a touch-and-hold for entering Chaudhri's reconfiguration mode.

2. Martyn's other methods of calling up the App Snapshot disclose further options avoiding conflict with Chaudhri

136. Further emphasizing the illogical and unnecessary nature of Petitioner's insistence of adopting a call up method not found in the art and that conflicts with Chaudhri's existing functionality, applying Martyn's other disclosed call up

methods also avoids a conflict. As an example, Martyn discloses beginning with the same “conventional navigation steps” discussed in ¶130 above, and then “call[ing] or fir[ing]” the Snapshot by ceasing scrolling such that “the highlight rest[s] on the name of an application in the App Launcher for a certain amount of time” until a “timeout” is reached. EX1005, [0016]-[0017]. A POSITA would apply this method to Chaudhri using the same “rotating the click wheel” to “navigate among and interact with one or more graphical objects (e.g. icons) displayed in touch screen” as discussed above. EX1004, [0093]. This method accurately follows Martyn’s disclosure and does not result in a conflict.

137. Critical to a timeout method is a highlight navigation. As Chaudhri describes, a click wheel allows a user to “navigate among and interact with one or more graphical objects (e.g. icons) displayed in touch screen.” EX1004, [0093]. A POSITA would appreciate that for a timeout method, there must be some indication to the user which application will be triggered when performing no additional actions. Without a visual highlight or the ability to use some indication for which application a user desires to perform operations for, timing out would frustrate a user as a user would have no control over selecting the application. For example, a timeout calling method without a highlight would continuously perform operations on an application but the user does not know which application until after the timeout

has been triggered. In this way the click wheel provides the requisite highlight navigation.

138. As another example, a POSITA would have created multiple alternative non-conflicting call up methods using one of Martyn’s “[o]ther selection processes”: voice activation. EX1005, [0017].

139. Voice activation offers similar benefits of calling Martyn’s Snapshot directly from the normal mode. Voice activation offers the benefit of a user directly opening Martyn’s Snapshot without having to place a finger on a touch-screen thereby reducing the number of steps and time needed to reach a desired action or shortcut.

140. Like Martyn, Chaudhri also discloses “accept[ing] verbal input for activation or deactivation of some functions through microphone 113.” EX1004, [0085]. Thus, Martyn’s disclosure of voice activation can be directly applied to Chaudhri. As Dr. Cockburn explained, there “would be more than one way to complete voice activation of [Martyn’s] Snapshot,” none of which would create a conflict with an already assigned input method. EX 2007, 34:2-6. “[I]n a two part process” a user could state “Messages. Show Snapshot,” with a pause between stating “messages” and “show Snapshot.” *Id.*, 35:6-9. Or “in a one-part process,” a user could state “Show message Snapshot” without a pause. *Id.*

141. Each of these methods may be readily applied to Chaudhri to call up an App Snapshot without creating a conflict with Chaudhri's existing functionality. Voice activation does include a user placing their finger at the location of an application icon as is required for entering Chaudhri's reconfiguration mode. Therefore, there is no conflict between a voice activation calling method for Martyn's Snapshot and entering Chaudhri's reconfiguration mode.

F. A Self-Induced Conflict is the Only Motivation for Adding Griffin.

142. Petitioner argues that a "POSITA would have looked to incorporate Griffin's teachings to evaluate a duration parameter of the user's touch against certain predetermined criteria" because a "POSITA would have [needed to] [seek] a solution to the problem of distinguishing between [] different gesture commands" of "[1] a tap to launch the application, [2] a touch-and-hold to trigger a contextual sub-menu, and [3] another touch-and hold to transition into the reconfiguration mode." Pet., 21. But Petitioner ignores that a POSITA only needed to seek Griffin, because Petitioner created a conflict by taking a specific calling method from Martyn. Pet., 16. Chaudhri already teaches what Griffin allegedly supplies—"evaluat[ing] a duration parameter of the user's touch against certain predetermined criteria." Pet., 16; EX1004, [0217] ("the device detects a contact ... for more than a predetermined period of time"), EX1004, [0153] ("receiving an input ... for more than a predetermined duration").

143. Chaudhri discloses “[f]or example, in FIG. 5A, the device detects a contact 5008 with the solitaire application icon 5002-4 for more than a predetermined period of time (e.g., 2 seconds) and, in response, the device enters a user interface reconfiguration mode, as illustrated in FIG. 5B.” EX1004, [0217]. This disclosure specifically mentions a “predetermined period of time (e.g., 2 second).” *Id.* For Chaudhri’s device to detect a length of contact it must be able to evaluate a duration of a touch.

144. Similarly, Chaudhri discloses “[i]n other embodiments, peripheral interface 118 transmits event information only when there is a significant event (e.g., receiving an input above a predetermined noise threshold and/or for more than a predetermined duration).” EX1004, [0153]. In accordance with this disclosure, considering the significance an event requires a device cable of detecting a touch of varying durations and hence for different time thresholds.

145. Absent creating a conflict there is no element for Griffin to supply. Therefore, Petitioner’s motivation is not based in the references or the knowledge of a POSITA but rather uses the claims as a map to reconstruct the invention.

XII. Petitioner fails to properly map the required claim elements in multiple dependent claims

146. Petitioner asserts that “[t]he Chaudhri-Martyn-Griffin Combination renders obvious” Limitations [21], [22], [35], [47], and [57]. Pet., 43-44, 51-52, 59-60. As described below Petitioner fails to find each of these Limitations in “[t]he

Chaudhri-Martyn-Griffin Combination” and therefore fails to render any of these Limitations obvious.

A. [21] and [22]: Content Indication

147. Claim 21 recites “[t]he electronic device of claim 1, wherein: the first application is a content creation application; the one or more action option objects include a respective action option object that is associated with displaying at least a portion of previously created content; and wherein the one or more programs include instructions for: detecting selection of the respective action option object, and, in response to detecting selection of the respective action option object, *displaying the at least portion of previously created content.*” Claim 21 depends on claim 22 and recites “wherein the content creation application includes an email application, and *the displayed at a least portion of previously created content* includes a previous email message, for generating a reply thereto.

148. Petitioner maps “displaying the at least portion of previously created content” of claim 21 to “unread emails” and “Chat ongoing” shown in Figure 3 of Martyn reproduced below. Pet., 43-44. Similarly, Petitioner maps “the displayed at least portion of previously created content” of claim 22 to the same “unread emails” indication. Pet., 44.

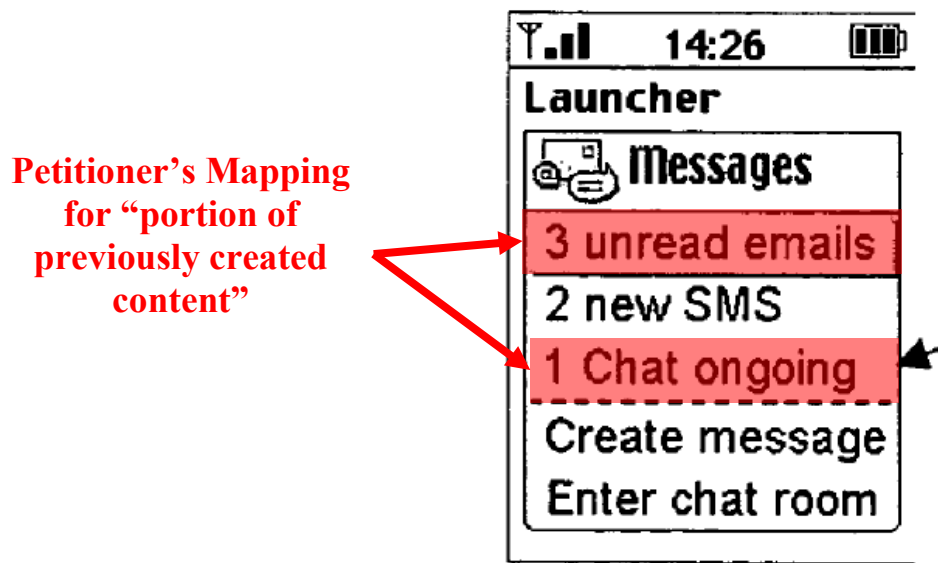


Figure 3

EX1004, Figure 3

149. Neither “unread emails” nor “Chat ongoing” *display* any *portion of previously created content* as required by the claims. At best Martyn discloses indications informing a user that previously created content is available for viewing. As seen above, no content of the “unread emails” or “chats” are displayed.

150. An unread email is an email that has not been opened or has been marked as such by a user. An email can be previously created content. The content of the email can include the body of that email or the fields (e.g. To:, From:, etc.). A quantity of unread emails and the category itself does not show any of that content. Therefore, Martyn’s Figure 3 does not display any portion of previously created content.

151. This is similarly true for “chat ongoing.” A chat ongoing may refer to an exchange of messages. In the exchange of messages, a first user would see information sent by a second user such that they could respond to that message. The message sent by the second user is previously created content. The description that a chat is ongoing or the ability to access these exchanged messages do not show the content of the messages themselves. Therefore, Martyn’s Figure 3 does not display any portion of previously created content.

B. [35]: Multiple Moved Apps

152. Claim 35 recites “[t]he electronic device of claim 1, wherein the electronic device is configured such that, in accordance with the determination that the first single-finger touch input meets the one or more movement criteria, multiple of the plurality of application icons is moved in the foreground virtual display layer so that the multiple of the plurality of application icons appear to float above the background virtual display layer.

153. Petitioner maps “multiple of the plurality of application icons is moved” “in accordance with the determination that the first single-finger touch input meets the one or more movement criteria” to Chaudhri’s disclosure that “the remaining *plurality of application icons* continue to jiggle (*move*) as the first application icon moves across the display” while in Chaudhri’s reconfiguration mode. Pet., 51.

154. Petitioner’s mapping ignores the claim language. The claim requires that multiple application icons are moved *d* when a movement criteria is satisfied. The application icons that Petitioner attempts to map to are *already* jiggling because Chaudhri’s device is in reconfiguration mode, which petitioner previously mapped to performing an operation when an *operation criteria is satisfied*. Pet., 34 Therefore, the application icons that Petitioner points to for this mapping are not moved when a movement criteria is satisfied.

155. Chaudhri describes “user interface objects jiggle” once “the device has entered a user interface configuration mode.” EX1004, [0218]. This discloses to a POSITA that the jiggling of applications begins after the device has entered reconfiguration mode and they continue to jiggle while in reconfiguration mode. This occurs before any relocation of an application icon. EX1004, [0220]. Chaudhri describes that a touch and subsequent movement of a racing icon is depicted in FIG 5B and that “[i]n FIG. 5B, the device has entered a user interface reconfiguration mode.” EX1004, [0218], [0220]. Therefore, the jiggling of the applications do not begin in response to a movement criteria (touch and subsequent movement) but rather when Chaudhri’s device enters reconfiguration mode.

C. [47]: Interactive User Element

156. Claim 47 recites “[t]he electronic device of claim 1, wherein the electronic device is configured such that, during detection of the first single-finger

touch input and prior to displaying the one or more action option objects, *a user interface element is displayed to appear at least partially behind the first application icon.*”

157. Petitioner maps “user interface element” to Chaudhri’s background wallpaper. To support this mapping, Petitioner references a previous section describing Chaudhri’s background wallpaper as “providing the backdrop” for Chaudhri’s applications. Pet., 8; *see also* EX1004, [0268], [0366], [0368]. Petitioner also references its mapping for element [1.1], in this mapping Petitioner describes this background wallpaper as “static.” Pet., 37. The plain and ordinary meaning of a “user interface element” is interactive. Therefore, petitioner mapping to a non-interactive element cannot meet the requirements of the Limitation.

158. As discussed above in ¶¶55-58, the plain and ordinary meaning of user interface element requires more than an element that provides visual information to a user.

D. [57]: Dynamically Generated Content

159. Claim 57 recites “[t]he electronic device of claim 1, wherein the electronic device is configured such that at least one of the one or more *action option objects includes content that is dynamically generated.*” Petitioner maps the one or more action objects that include content that is dynamically generated to the “‘common functions’ of the associated application[s]” in Martyn’s Snapshot. Pet.,

59-60. Petitioner reasons that these “common functions” dynamically change because Martyn’s device learns over time which functions a user is interested in and accordingly varies what is shown to the user in Martyn’s Snapshot. Pet., 60. Petitioner ignores that the claim requires that the one or more action objects includes content that is dynamically *generated*, not that the action option objects are in some way “dynamic” as stated in Petitioner’s mapping. Pet., 60. Petitioner’s mapping only applies to the *action option objects* being dynamically rearranged. Nothing in Petitioner’s mapping points to any action option objects including dynamically generated content.

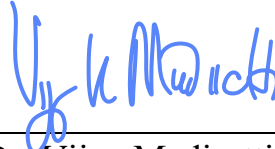
160. Martyn’s device adapts by assessing the frequency of functionality and stored data type use and adjusting which are displayed in Martyn’s Snapshot accordingly, but this does not equate to dynamically generating new content. EX1004, [0026]. Dynamic generation implies the creation of new content, whereas Martyn’s merely selects from already existing content. *Id.* Martyn’s device is able to learn because of existing “functionalit[ies] and/or stored data types.” *Id.* Because this learning is based on existing information already present in the device it does not create new functionalities or new data and cannot be dynamic generation.

XIII. CONCLUSION

161. 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,



Dr. Vijay Madiseti

Dated: April 15, 2025